A Comprehensive Survey of Frequent Itemsets Mining on Transactional Database with Weighted Items

Huan Phan[1,3], Bac Le[2,3]
[1]Faculty of Mathematics and Computer Science, VNUHCM-University of Science, Vietnam
[2]Faculty of Information Technology, VNUHCM-University of Science, Vietnam
[3]Vietnam National University, Ho Chi Minh City, Vietnam
Correspondence: Huan Phan, huanphan@hcmussh.edu.vn
Communication: received 5 April, revised 19 May, accepted 31 May
Digital Object Identifier: 10.32913/mic-ict-research.v2021.n2.967

***Abstract*: In 1993, Agrawal et al. proposed the first algorithm for mining traditional frequent itemset on binary transactional database with unweighted items - This algorithm is essential in finding hindden relationships among items in your data. Until 1998, with the development of various types of transactional database - some researchers have proposed a frequent itemsets mining algorithms on transactional database with weighted items (the importance/meaning/value of items is different) - It provides more pieces of knowledge than traditional frequent itemsets mining. In this article, the authors present a survey of frequent itemsets mining algorithms on transactional database with weighted items over the past twenty years. This research helps researchers to choose the right technical solution when it comes to scale up in big data mining. Finally, the authors give their recommendations and directions for their future research.**

Keywords: *Data mining, frequent itemsets, weighted items*

I. INTRODUCTION

Association rule mining is an important technique in data mining. The goal of mining is to discover relationships between data values in a dataset. The first model of association rule mining is the binary model, which is also known as the fundamental model, was proposed in 1993 by Agrawal et al. [1] to analyse data transaction, detect relationships between items sold in supermarkets. From there, they can have a reasonable investment and business plan, and organize the sales counter to have revenue in the most profitable trading sessions. In addition, this knowledge can be applied to predict the number of upcoming best-selling items and customer shopping trends. Synthesizing this knowledge to plan operations, production and business in a more convenient way will reduce the time for statistics, market research, etc.

The proposed algorithms for association rule mining are divided into two phases:

Phase 1: Find all itemsets that satisfy the minimum support threshold *minsup*. This phase takes a lot of time to process.

Phase 2: Generate association rules in turn from itemsets that satisfy *minsup* in phase 1 and these association rules must satisfy minimum confidence threshold *minconf*.

Agrawal has proposed the Apriori algorithm [2] - Frequent Itemsets (FI) mining, an algorithm that scans the data many times and has an exponential complexity. To improve the time in frequent itemsets mining, many researchers have proposed an efficient frequent itemsets mining algorithm based on the storage structures that reduce the search space such as SE-Tree [8], Prefix-Tree [4], IT-Tree [6], FP-Tree [3], etc. However, in practice, generating the frequent itemsets is time consuming and has a very large number of itemsets. Therefore, some researchers have proposed mining the Closed Frequent Itemsets (CFI) with fewer frequent itemsets such as A-Close [5], Charm [6], etc. Meanwhile, some other scientists also proposed mining the Maximal Frequent Itemsets (MFI) such as Pincer-Search [7], Max-Miner [8], etc. In some practical applications, to apply FI, CFI and MFI costs a lot of computation as well as a large number of frequent itemsets. Bayardo proposed mining the Maximum Length Frequent Itemsets (LFI) which is a subset of the frequent itemsets FI and contains only frequent itemsets of maximum length like the DepthProject [9], MAXLEN-FI [10], etc.

In the last years of the 20th century, along with the development of diverse transaction data and inheritance from Agrawal's traditional frequent itemsets mining; Ramkumart

et al. have proposed the WIS algorithm [11] that mines the frequent itemsets with weighted items (the importance / significance level of the items is different) containing more knowledge than the traditional frequent itemsets mining (without weighted of items). Besides, Cai et al. also proposed MINWAL algorithm [12] to solve the above problem. After that, many authors researched and proposed algorithms [13-21] to solve this problem. However, all algorithms approach and solve in the direction of satisfying the "downward closure property". In 2011, Huai et al. proposed the WHIUA algorithm [22] which, following the Apriori approach and "not satisfy the downward closure property", significantly increases the search space - this is a big challenge for researchers of data mining. In the next decade, a number of algorithms were proposed such as IWFP [23], PWA [27], WAC [30], etc. but most of them still solve the problem in the direction of satisfying the "downward closure property" and the algorithm are similar to traditional frequent itemsets mining. Therefore, it has motivated the authors to research and summarize a number of algorithms for mining frequent itemsets on weighted transaction data proposed in the period 1998 to present.

In Part II, the article presents the basic concepts of association rule mining, frequent itemsets, Apriori algorithm and common data structures. In Part III, we summarizes some algorithms for mining frequent itemsets on transaction database with weighted items over the years. Discussion and recommendations are presented in Part IV; Conclusions and development directions are presented in Part V.

II. THE BASIC CONCEPTS

1. Mining association rules

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of m properties, each property is called an item. The set $W = \{w_1, w_2, \ldots, w_m\}, \forall w_j \geq 0$ is the weight (significance/importance) of each item. The set of items $X = \{i_1, i_2, \ldots, i_k\}$ $\forall i_j \in I\,(1 \leq j \leq k)$ is called itemset, itemset with k items is called k -itemset. $\mathcal{D}$ is transaction database, including n distinct records called set of transaction $T = \{t_1,\ t_{2,\ldots,}t_n\}$, each transaction $t_k = \{q_1 i_{k_1}, q_2 i_{k_2}, \ldots, q_m i_{k_m}\}$, $i_{k_j} \in I\,(1 \leq k_j \leq m)$ and $q_j \geq 0$ is the quantity/probability of item i_{kj} in transaction t_k.

Definition 1: Let $X, Y \subseteq I$ with $X \cap Y = \varnothing$, the association rule is a implication of the form $X \rightarrow Y$, satisfying two given thresholds (minsup - minimum support; minconf - minimum confidence), where X is called the antecedent and Y is the consequent.

Definition 2: The support of itemset $X \subseteq I$, denoted $\sup(X)$− the ratio between the number of transactions in $\mathcal{D}$ containing X and n transactions.

$$\sup(X) = \frac{|\{t \in D \mid X \subseteq t\}|}{n} \tag{1}$$

Definition 3: The support of the association rule $X \rightarrow Y$, denoted $\sup(X \rightarrow Y)$-the ratio between the number of transactions in $\mathcal{D}$ containing $\{X \cup Y\}$ and n transactions.

$$\sup(X \rightarrow Y) = \sup(X \cup Y) \tag{2}$$

Definition 4: Confidence of association rule $X \rightarrow Y$, denoted $\mathrm{con}\,f(X \rightarrow Y)$ - the ratio between the number of transactions containing $\{X \cup Y\}$ and the number of transactions containing X in $\mathcal{D}$.

$$\mathrm{conf}(X \rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)} \tag{3}$$

When we say that the confidence of a rule is 95%, it means that 95% of the transactions that contain the antecedent X also contain the consequent Y - "the conditional probability that the event Y occurs is 95%", with the condition "event X occurs". The confidence of the association rule represents the correlation between the events X and Y. The confidence is used to measure the significance of the rule. Usually in tasks, users are only interested in highl confidence association rules.

TABLE I
TRANSACTION DATABASES CLASSIFICATION

Data type	Feature description
Binary (traditional)	$\forall w_j = 1, \forall q_j = 1\ (1 \leq j \leq m)$
Uncertain (Fuzzy)	$\forall w_j, q_j \in [0,1]\ (1 \leq j \leq m)$
Unweighted quantity and item	$\forall w_j = 1, \forall q_j \geq 0\ (1 \leq j \leq m)$
Weighted quantity and item	$\forall w_j \in [0,1], \forall q_j \geq 0\ (1 \leq j \leq m)$
Weighted item*	$\forall w_j \in [0,1], \forall q_j = 1\ (1 \leq j \leq m)$

(*) In this article, the authors only focus on surveying the research related to the frequent itemsets on transaction database with weighted items.

Pseudocode 1: Association Rules Mining

Input: Transaction database $\mathcal{D}$, minsup, minconf

Output: Association Rules

1. $FI = \{\varnothing\}$ //frequent itemset − Phase 1
2. For each $Z \in P_{\geq 1}(I)$ do
3. If $\sup(Z) \geq$ minsup then
4. $FI = FI \cup Z$
5. $AR = \{\varnothing\}$ //Association Rules − Phase 2
6. For each $fi \in FI$ do
7. If $\mathrm{con}\,f(X \rightarrow \{fi \setminus X\}) \geq$ minconf then
8. $AR = AR \cup (X \rightarrow \{fi \setminus X\})$
9. Return AR

($P_{\geq 1}(I)$: all powerset of I have at least one item)

Algorithm 1 is divided into 2 phases

Phase 1: (Lines 1 to 4) this is the phase to find all itemset that satisfy the minsup threshold;

Phase 2: (Lines 5 to 8) generate association rules from itemset that satisfy minsup in phase 1 and these association rules must satisfy minconf (association rule $X \rightarrow \{fi \setminus X\}$ has the antecedent X and the consequent $Y = \{fi \setminus X\}$ satisfying $X \cap Y = \varnothing$).

Analyze space search algorithm 1:

Phase 1: transaction database $\mathcal{D}$ has $|I| = m$, all itemset generated from I have at least one item - $P_{\geq 1}(I)$, we have a space of combinations between items generated from m items of $2^m - 1$, this is search space in phase 1 for identifying itemsets that satisfy the minsup threshold.

Phase 2: after determining the set of combinations or itemsets satisfying minsup, this phase generates association rules from the itemsets in turn. Suppose, for itemset X with m items, $l(1 \leq l \leq m)$ and $r(1 \leq r \leq m-l)$ are the number of items in the precondition and the conclusion. Then, we have C_l^m number of ways to choose the antecedent with l items over m items from itemsets X and C_r^{m-l} number of ways to choose the consequent with r items on $(m-l)$ items left from itemset X:

$$\sum_{l}^{m} \sum_{r}^{m-l} C_l^m C_r^{m-l} \tag{4}$$

Applying Newton's triangle, equation (4) is rewritten as follows:

$$\sum_{l}^{m} \sum_{r}^{m-l} C_l^m C_r^{m-l} = 3^m - 2^{m+1} + 1 \tag{5}$$

Most of the proposed association rule mining algorithms focus on the improvement technique in phase 1 (determination of itemsets satisfying minsup). All these algorithms assume that phase 1 takes the most time in the whole association rule mining process. However, according to the above analysis, phase 2 is more complex and has a large generation space; or in the other words, in algorithm 1, every phase is necessary to improve and enhance calculating performance. For example, we have itemset $X = \{A, B, C\}, m = 3$, the total number of association rules generated from itemset X as calculated by equation (5) is $C_1^3 \times \left(C_2^2 + C_1^2\right) + C_2^3 \times \left(C_1^1\right) = 3 \times (1+2) + 3 \times (1) = 12$, that means for itemset X have 3 items, it is necessary to consider whether the 12 association rules satisfy the minconf threshold or not. Similarly, if itemset X has 4 items $(m = 4)$, then we need to consider 50 association rules that satisfy minconf. In addition, the above analysis also clearly shows that the search space in both phases does not depend on the number of transactions of the dataset D but only on the number of items of the dataset mining.

TABLE II
ESTIMATING THE GENERATION SPACE OF COMBINATIONS AND THE NUMBER OF ASSOCIATION RULES GENERATED FROM M-ITEMSET

m	10	20	30	40	50
P_1	$2^{10} - 1$	$2^{20} - 1$	$2^{30} - 1$	$2^{40} - 1$	$2^{50} - 1$
P_2	$3^{10} - 2^{11} + 1$	$3^{20} - 2^{21} + 1$	$3^{30} - 2^{31} + 1$	$3^{40} - 2^{41} + 1$	$3^{50} - 2^{51} + 1$
P_2/P_1	57	3.325	191×10^3	11×10^6	637×10^6

P_1: the space for generating combinations $(2^m - 1)$ in Phase 1;
P_2 : the number of association rules generated from m - itemset $\left(3^m - 2^{m+1} + 1\right)$ in Phase 2.

Table II, shows that the ratio between the space generated in Phase 2 and Phase 1 is very large when the number of items are increased, that is Phase 2 has a very large space compared to Phase 1.

Definition 5: Let $X \subseteq I, X$ is called the frequent Denoted FI is the set of the frequent itemsets.

Definition 6: Let $X \subseteq I, X$ is called the Closed frequent itemset – if X is a frequent itemset and there is no parent set of the same support. Denoted CFI is the set of closed frequent item sets.

Definition 7: Let $X \subseteq I, X$ is called the maximal frequent itemset - if X is a frequent itemset and there is no parent set of frequent itemsets. Denoted MFI is the set of the maximal frequent itemsets.

In addition, when it is necessary to mine the association rules with the largest number of items, the researchers also propose mining the frequent itemset of maximum length - which is a subset of the maximal frequent itemset and has the maximum length (**LFI** ⊆ **MFI** ⊆ **CFI** ⊆ **FI**).

Definition 8: Let $X \in$ **FI**, X is called the frequent itemset of maximum length - if $\forall Y \in$ **FI** then $|X| \geq |Y|$, that means, the number items of itemset X greater than or equal the number items of any frequent itemsets contained in FI. Denoted LFI is the set of the maximum length frequent itemsets.

Some properties of frequent itemsets: these are the fundamental properties used for reducing the search space - these properties are called the Downward Closure Property (DCP), which is also known as Apriori properties.
Property 1: $\forall X \subseteq Y : \sup(X) \geq \sup(Y)$;
Property 2: $\forall X \subseteq Y, \sup(Y) \geq \text{minsup} : \sup(X) \geq \text{minsup}$;
Property 3: $\forall X \subset Y, \sup(X) < \text{minsup} : \sup(Y) < \text{minsup}$.

2. Apriori algorithm

The algorithm proposed by Agrawal in 1994 [2], is considered historic in the association rule mining, because it is far beyond the reach of familiar algorithms. Apriori is the foundational algorithm for finding frequent itemsets using

candidate generation methods. The algorithm is characterized by Breadth-First Search using the Apriori property: any infrequent $(k-1)$-itemset cannot be a sub-itemset of the frequent k-itemsets.

Pseudocode 2: Frequent Itemset Mining
Input: Transaction database $\mathcal{D}$, minsup
Output: Frequent Itemsets

1. Generate candidate C_{k+1} from frequent k-itemset;
2. Scan data - calculate support of candidates;
3. Add the candidates to the $(k+1)$-itemset list.

Advantage: The algorithm is based on a fairly simple comment that any sub-itemset of frequent itemsets are also frequent itemsets (property 2). Therefore, in the process of finding the candidate itemset, the algorithm only needs to use the candidate itemset that appeared in the previous step, not all of the candidate itemsets (up to that point). So, the memory is significantly released.

Disadvantage: The algorithm has to scan the data (max +1) times with max being the length of the longest frequent itemset. The Apriori algorithm reduces the space based on the Apriori property. However, when the number of frequent itemsets generated is large, the fact that the max is large or the minsup is small will result in generating a lot of candidate itemsets and having to traverse the data many times, the algorithm has a high cost. In practice, 2^{100} candidates (in the worst case) need to be generated to find the frequent itemset of size 100.

From the above disadvantages, many researchers have proposed algorithms to increase the performance of generating frequent itemsets based on the organization of data storage and corresponding effective search strategies.

3. The common data structure on mining the traditional frequent itemsets

In addition to some algorithms with Apriori approach, we present a survey of common data structures used by many authors for storing the search space in the mining of frequent itemset and search strategies along with storage space. Search strategies on tree structure: Depth First Search - DFS, Breadth First Search - BFS or a combination of both.

a) Lattice

This is a data structure that is used a lot in the proposed algorithms. Transaction database D have $|I| = m$, all itemset (powerset) are generated from I $P(I)$, we have the space of combinations between items generated from m items is 2^m, this is the full search space. In 1999, Pasquier proposed the A-Close algorithm [5] based on the Lattice structure to mining the closed frequent itemset (nearly 2,100 citations).

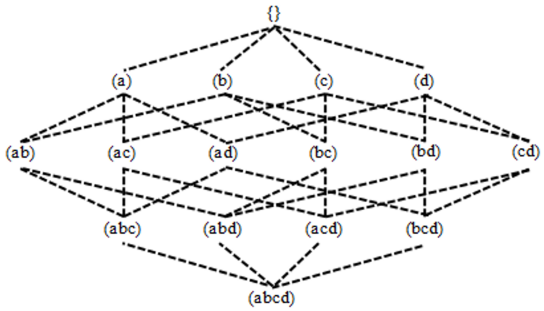


Figure 1. Lattice

Search strategy: combining both dimensions (Top-Down and Bottom-Up).
Advantages: simple, ensure mining in full the frequent itemsets.
Disadvantages: takes a lot of memory during mining.

b) Set Enumeration Tree (SE-Tree)

In 1998, Bayardo proposed the Max-Miner algorithm [8] based on an enumeration tree (over 2,000 citations) to mine the maximal frequent itemsets.

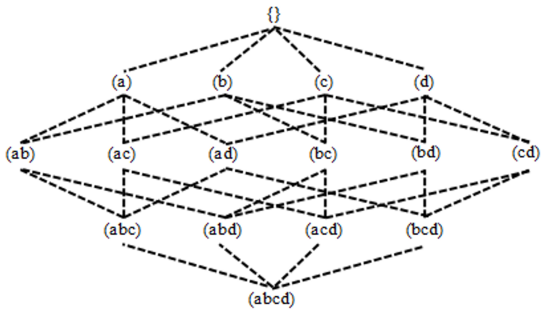


Figure 2. Set Enumeration Tree (SE-Tree)

Search strategy: DFS.
Advantages: simple, ensure mining in full the frequent itemsets.
Disadvantages: unbalance in the mining process (depending on the frequency of the item).

c) Prefix-Tree

In 2002, Liu proposed the OpportuneProject algorithm [4] based on the Prefix-Tree structure (nearly 300 citations) to quickly mining the frequent itemset. Similar to an SE-Tree, the contents of the archive are shortened through prefixes.

d) IT-Tree (Itemset Tidset Tree)

In 1999, Zaki proposed the Charm algorithm [6] along with an IT-Tree (Itemset Tidset Tree) structure like a list tree

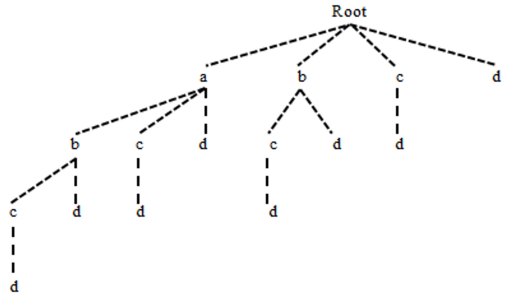


Figure 3. Prefix-Tree

structure and a combination of storing transaction identifier of itemset (nearly 1,600 citations).

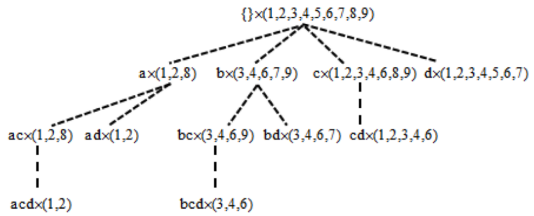


Figure 4. IT-Tree

Search strategy: DFS.
Advantages: simple, ensure mining in full the frequent itemsets.
Disadvantages: unbalance during mining and memory intensive storage.

e) FP-Tree (Frequent Pattern Tree)

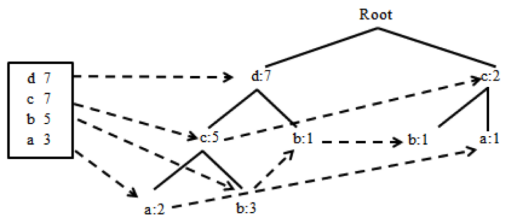


Figure 5. FP-Tree

In 2000, Jiawei Han et al. proposed the FP-Tree structure [3] (nearly 9,000 citations) and the FP-Growth algorithm for mining frequent itemsets. This is an extended tree from Prefix-Tree, with an additional header table that stores items, frequencies and links to nodes in the tree. From that, the FP-Tree structures have been used and improved by many researchers.

Although the FP-Tree structure is more reduced than the above ones', FP-Tree is an extension from Prefix-Tree, so there are still many limitations similar to Prefix-Tree - unbalanced tree.

III. SURVEY OF FREQUENT ITEMSETS MINING ON TRANSACTIONAL DATABASE WITH WEIGHTED ITEMS

1. Algorithm grouping

In practice, the properties in the data are not equal. There are some properties that are emphasized, we say those properties have a higher importance level than others. Over the past 20 years, this has always been a very interesting research and many proposalsed have been made (by researchers) to solve this problem. In this survey, the authors categorize as well as group the research works of domestic/foreign researchers based on the method of calculating the measures related to the weights of the itemset.

a) Some foreign research work

Group 1: In 1998, G. D. Ramkumar et al. proposed the WIS algorithm [11].

Description WIS algorithm (1998)	
Apriori property	Satisfy the downward closure property
Weighted items	Weight of each item $w_j \in [0,1]$
Approach	Apriori
Association rule	Weighted binary
Transaction weight	$tw(t_k) = \frac{\sum_{i_j \in t_k} w_j}{\lvert t_k \rvert}$
Weighted support	$ws(X) = \frac{\sum_{t_k \in t(X)} tw(t_k)}{\sum_{t_k \in T} tw(t_k)}$

In 2003, Feng Tao et al. proposed the WARM algorithm [13]. The proposed algorithm inherits the calculation method of G. D. Ramkumar and extends the unbound weighted itemset belong to $[0,1]$. To speed up the computation, Feng Tao uses the Lattice structure, which is a structure widely used in mining data without weights.

Description WARM algorithm (2003)	
Apriori property	Satisfy the downward closure property
Weighted items	Weight of each item w_j (unbound $\in [0,1]$)
Approach	Lattice
Association rule	Weighted binary
Transaction weight	$tw(t_k) = \frac{\sum_{i_j \in t_k} w_j}{\lvert t_k \rvert}$
Weighted support	$ws(X) = \frac{\sum_{t_k \in t(X)} tw(t_k)}{\sum_{t_k \in T} tw(t_k)}$

Group 2: In 1998, Chun Hing Cai et al. proposed the MINWAL(O) and MINWAL(W) algorithm [12].

Description MINWAL algorithm (1998)	
Apriori property	Satisfy the downward closure property
Weighted items	Weight of each item $w_j \in [0,1]$
Approach	Apriori-like
Association rule	Weighted binary
Weighted itemset	$w(X) = \frac{1}{\lvert X\rvert}\left(\sum_{i_j \in X} w_j\right)$
Weighted support	$ws(X) = w(X) \times \sup(X)$

In 2005, Unil Yun et al. proposed the WFIM algorithm to inherit the calculation method from Chun Hing Cai. From 2005 to now, Unil Yun's team has more than 20 research works related to weighted association rule mining. However, these works are all based on the WFIM algorithm [14].

Description WFIM algorithm (2005)	
Apriori property	Satisfy the downward closure property
Weighted items	Weight of each item w_j (unbound ϵ $[0,1]$)
Approach	FP-Tree (2005), Prefix-Tree (2012)
Association rule	Weighted binary
Weighted itemset	$w(X) = \frac{1}{\lvert X\rvert}\left(\sum_{i_j \in X} w_j\right)$
Weighted support	$ws(X) = w(X) \times \sup(X)$

Group 3: In 2011, Zi-guo Huai et al. proposed the WHIUA algorithm [22]. Although nearly a decade has passed, Zi-guo Huai's work has only two citations. According to the author's survey, this is a challenging approach in mining association rule with weighted - the WHIUA algorithm approaches in the direction of NOT satisfying the downward closure property, leading to a very large search space and is not suitable for common pruning strategies.

Description WHIUA algorithm (2011)	
Apriori property	NOT Satisfying the downward closure property
Weighted items	Weight of each item $w_j \in [0,1]$
Approach	Apriori combines hash table
Association rule	Weighted binary
Weighted itemset	$w(X) = \mathrm{Max}_{i_j \in X}\, w\left(i_j\right)$
Weighted support	$w\sup(X) = w(X) \times \sup(X)$

Group 4: In 2013, Guo-Cheng Lan et al. proposed the PWA algorithm [27]. In order to implement effective pruning strategies and satisfy the downward closure property, the author proposed the upper bound weights of the transactions as well as the itemset.

Description WFIM algorithm PWA (2013)	
Apriori property	NOT Satisfying the downward closure property
Weighted items	Weight of each item $w_j \in [0,1]$
Approach	Apriori-like
Association rule	Weighted binary
Weighted itemset	$w(X) = \frac{1}{\lvert X\rvert}\left(\sum_{i_j \in X} w_j\right)$
Weighted support	$ws(X) = \sum_{X \in t_k, t_k \subset TDB} w(X)$
Transaction-weight upper-bounds	$\mathrm{twub}\,(t_k) = \mathrm{Max}_{i_j \in t_w} w\left(w_{i_j}\right)$
Weighted frequent upper-bound itemset	$\mathrm{wsub}\,(X) = \sum_{X \in t_k, t_k \subset TDB} \mathrm{twub}\,(t_k)$

Group 5: In 2015, Xuyang Wei et al. proposed the IWFPM algorithm [34]. In order to implement effective pruning strategies and satisfy the downward closure property, the author proposed the upper bound weights of the transactions as well as the itemset.

Description IWFPM algorithm (2015)	
Apriori property	NOT Satisfying the downward closure property
Weighted items	Weight of each item $w_j \in [0,1]$
Approach	Apriori
Association rule	Weighted binary
Weighted itemset	$w(X) = \frac{\prod_{i_j \in X} w(i_j)}{\sum_{i_j \in X} w(i_j)}$
Weighted support	$w\sup(X) = \sup(X) \times \frac{\prod_{i_j \in X} w(i_j)}{\sum_{i_j \in X} w(i_j)}$

In addition, there has been a lot of research related to the mining of weighted association rules in recent years - these algorithms, mostly using the above calculation methods (5 groups) belong with data structures and appropriate pruning strategy or adding constraints.

2. Some domestic research work

Through a survey of the domestic research on mining the weighted frequent itemset of items, there is a prominent research group of Le Hoai Bac and Vo Dinh Bay. From 2009 to now, the above research group has had related works which are summarized as the following table:

Description WIT-FWIs algorithm (2010)	
Apriori property	Satisfy the downward closure property
Weighted items	Weight of each item $w_j \in [0,1]$
Approach	IT-Tree (2010, 2013, 2017), DBV (2016), Prefix-Tree (2016, 2018, 2020, 2021)
Association rule	Weighted binary
Transaction weight	$tw\,(t_k) = \frac{\sum_{i_j \in t_k} w_j}{\lvert t_k\rvert}$
Weighted support	$ws(X) = \frac{\sum_{t_k \in t(X)} tw\,(t_k)}{\sum_{t_k \in T} tw\,(t_k)}$

The research team uses the method to calculate weighted based on group 1 (G.D.Ramkumar, 1998) and use data structures with the appropriate pruning strategies to increase mining efficiency.

In addition, the domestic research groups are interested in research in the mining data number or high-interested samples and there are not many works related to the research problem. Therefore, the authors only focus on surveying the prominent group above.

3. Some algoritms for mining the frequent itemsets with weighted items

Synthesizing some of the frequent itemsets mining works in transaction database with weighted items by time and classified in 5 groups.

Table III, presents the timeline of some frequent itemsets mining algorithm on the transaction database with weighted items from 1998 to present, including information fields: top author's name, algorithm's name, satisfy Apriori property, with the data structure approach, the number of citations of the work [11-50] (Cai [11] to Bui [50]), the year of publication and the group of algorithms.

TABLE III
SYNTHESIZING 40 FREQUENT ITEMSETS MINING WORKS WITH WEIGHTED ITEMS [11-50] (MARCH 2021)

First author	Algortithms	DCP	Approach	Cite	Yr	Group
Ramkumar	MINWAL[11]	✓	Apriori	645	1998	2
Cai	WIS[12]	✓	Apriori	84	1998	1
Tao	WARM[13]	✓	Lattice	487	2003	1
Yun	WFIM[14]	✓	FP-Tree	72	2005	2
Geng	GTCWFP[15]	✓	FP-Tree	4	2008	2
Ahmed	IWFP[16]	✓	FP-Tree	21	2008	2
Le	WIT-FWI[17]	✓	IT-Tree	10	2010	1
Jeong	DWFPM[18]	✓	FP-Tree	6	2010	2
Pears	EWGen[19]	✓	Apriori-like	6	2010	2
Wang	DWCI[20]	✓	FP-Tree	1	2011	1
Yun	WAF[21]	✓	FP-Tree	53	2011	2
Huai	WHIUA[22]	×	Apriori + Hash	5	2011	3
Ahmed	IWFP[23]	✓	FP-Tree	64	2012	2
Yun	MWFIM[24]	✓	lattice	46	2012	2
Vo	WITFWIs[25]	✓	IT-Tree	123	2013	1
Vo	FWCIs[26]	✓	IT-Tree	13	2013	1
Lan	PWA[27]	×	Apriori-like	9	2013	4
Yun	MCWP[28]	✓	FP-Tree	32	2013	2
Mohan	FWIDIFF[29]	✓	IT-Tree	1	2014	1
Yun	WAC[30]	✓	FP-Tree	20	2014	2
Nguyen	SWITD[31]	✓	IT-Tree	4	2015	1
Lan	PWAI[32]	×	Apriori-like	5	2015	4
Wei	IWFPM[33]	×	FP-Tree	5	2015	5
Lee	AWMFPs[34]	✓	FP-Tree	15	2016	2
Nguyen	IWSFWIs[35]	✓	IT-Tree + BDV	9	2016	1
Yun	IM_WMFI [36]	✓	FP-Tree	58	2016	2
Qin	WidTMWFIM[37]	✓	IT-Tree		2016	1
Yun	WFPmax[38]	✓	FP-Tree	13	2016	2
Lee	FWI[39]	✓	Prefix-Tree	17	2017	1
Lin	RWFIMEMine[40]	✓	SE-Tree	9	2017	2
Vo	WIT-FWCIDiff[41]	✓	IT-Tree	12	2017	1
Bui	NFWI[42]	✓	Prefix-Tree	4	2017	1
Kiran	WFRIM[43]	✓	FP-Tree		2017	2
Zhao	SWFP[44]	✓	FP-Tree	5	2018	2
Klangwisan	WFRIMIWS[45]	✓	FP-Tree		2018	2
Cengiz	Pre/PostWAR M[46]	✓	Lattice		2019	1 + 2
Dewan	CPTDW[47]	✓	Prefix-Tree		2019	2
Yue	ENSFWI[48]	✓	Prefix-Tree		2019	1
Vo	TFWIN+[49]	✓	Prefix-Tree	4	2020	1
Bui	NFWCI[50]	✓	Prefix-Tree		2021	1

(✓: satisfy Apriori property; ×: not satisfy Apriori property)

IV. DISCUSSION AND RECOMMENDATION

1. Data structure

The authors have investigated 40 works on the frequent itemset mining on transaction database with weighted items [11-50]. Some algorithms follow Apriori approach, while others use the common data structures. The below table is a statistics of algorithms using data structures by each group. Table IV, shows the most used FP-Tree data structure (40.00%), followed by IT-Tree data structure (20.00%) and approach Apriori (15.00%) - this is the basic approach to mining frequent itemsets with weighted items. In addition, the above table also shows that the researches also focus on groups 1 and 2 (90.00%).

TABLE IV
RATIO OF ALGORITHMS IN THE SYNTHESIZE USING THE COMMON DATA STRUCTURES FOR MINING FI WITH WEIGHTED ITEMS

Approach	Ratio of algorithm numbers in the synthesize by group (%)					Total (%)
	1	**2**	**3**	4	**5**	
Apriori	2.50	5.00	2.50	5.50		15.00
Lattice	7.50					7.50
SE-Tree		2.50				2.50
Prefix-Tree	12.50	2.50				15.00
IT-Tree	20.00					20.00
FP-Tree	2.50	35.00			2.50	40.00
Total (%)	45.00	45.00	2.50	5.50	2.50	100.00

a) Apriori property

In frequent itemsets mining on unweighted transaction data of items, this is the fundamental property for reducing the search space for the itemsets satisfying the user-specified frequency threshold minsup. However, the observations in Table III show that 90.00% (corresponding to 36 works in groups 1 and 2) algorithms use satisfying Apriori property in the process of pruning and reducing the space generated frequent itemset on transaction databases with weighted items; only 10.00% (corresponding to 4 works in groups 3, 4 and 5) of algorithms solve the problem of mining frequent itemsets in transaction databases with weighted items that does NOT satisfy the Apriori property (the downward closure property) - this is a big challenge in mining frequent itemsets because the search space is very large. This requires researchers to come up with the storage techniques as well as the strategies to reduce the search space.

In addition, the algorithms in groups 3, 4 and 5 approach in the direction of NOT satisfying the Apriori property (only 04/40 works) together with the proposed algorithm base the Apriori algorithm.

2. The method to calculate the weighted itemset

In this section, the authors discuss the measures to calculate the 5 groups of algorithms above: transaction weight, weighted itemset, weighted support, transaction-weight upper-bound , weighted frequent upper-bound itemset and support of interest in algorithms.

TABLE V
DESCRIPTION OF THE MEASURES IN THE CALCULATION METHOD OF THE ALGORITHM GROUP

The measures including the calculation method of each group of algorithms	Algorithm group				
	1	**2**	**3**	**4**	**5**
Transaction weight	✓				
Weighted itemset		✓	✓	✓	✓
Weighted support	✓	✓	✓	✓	✓
Support		✓	✓		✓
Transaction-weight upper-bound				✓	
Weighted frequent upper-bound itemset				✓	

Table V, shows the weights frequently calculated in all 5 groups of algorithms - this is the basic measure for

evaluating whether itemsets are frequent or not. Only group 1 considers transaction weight; group 4 proposes adding the transaction-weight upper-bound and the weighted frequent upper-bound itemset; The itemset weights are calculated in groups 2-4. In addition, the frequent itemsets are the support of groups 2,3 and 5 - this is an important measure to show the frquency of frequent itemsets appearance in transaction database. In groups 1 and 4, the support itemsets are not used - the frequent itemsets mined from these 2 groups are difficult to assess as frequent or not.

The weight itemset is calculated in groups 2-4: in groups 2 and 4, this measure is averaged according to the weight of items in itemsets - losing the significance of the weights of items in data transactions; In group 3, this measure is calculated according to dominance - that is, the weighted of itemset is largest weight of the items in the itemset (representing the significance of weight or the important of items in the transaction database).

In 2013, Lan et al. proposed PWA algorithm [27] - the first algorithm in group 4: adding a measure of the transaction-weight upper-bound and the weighted frequent upper-bound itemsets to help prune the search space faster.

In group 5, Wei et al. proposed IWFPM algorithm [34]. The algorithm for calculating the itemset weight is equal to the ratio between the product of the weighted items and sum of the weighted items in itemset - with this measure, the authors have not yet explained the meaning/role of the weights in the work.

In addition, basing on the method of calculating the measures of the 5 groups of algorithms, the authors have the following comments: from group 1 to 4, when changing to the traditional form of mining frequent itemset (the weighted items are the same) then the measures are corresponding - the weighted frequent becomes frequent; especially for group 5, the uncorrelated weighted frequent is the frequent when applied to the unweighted data transactions of items.

3. Recommendations

In the above section, the authors presented and discussed the frequent itemsets mining works on the weighted data transactions of items, which is surveyed from data structure, the search space pruning/reducing strategy and the method to calculate the relevant metrics in the mining process, the authors have the following recommendations:

– First, the algorithms for mining frequent itemsets on transaction database with weighted items, the experiment evaluation needed to be further compared with the traditional efficeint frequent itemsets mining algortihms (that means assign the weight of items to equal 1) to show that the proposed algorithm is really efficient;

– Second, it is necessary to choose a method to calculate the appropriate measures - the importance is that the measures represent the weighted role/meaning of each items, as well as the frequency of occurrence of the itemsets and especially the metrics to use that must be suitable when changing to unweighted one (or the weight of items is equal to 1);

– Third, according to the recommendations above - the authors propose to correct the calculation formula for group 5, specifically adding $|X|$ in weighting itemset;

Description IWFPM algorithm (2015)	
Weighted itemset	$w(X) = \lvert X\rvert \times \frac{\prod_{i_j \in X} w(i_j)}{\sum_{i_j \in X} w(i_j)}$
Weighted support	$w\,\sup(X) = \sup(X) \times w(X)$

– Fourth, researchers need to focus on proposing a frequent itemsets mining algorithm on transaction database with weighted items following the approach that does NOT satisfy the Apriori property - this is really a big challenge in this type of problem. Currently, the proposed research works only approach the Apriori algorithm and have no effective algorithm to solve this problem;

In addition, the authors also found that the approach in mining frequent itemsets on transaction database with weighted items is also based on the aggregated data structures used in traditional frequent itemsets mining. Then, the efficient algorithms in traditional mining frequent itemsets can be used for group 1 and 2 (satisfy Apriori property) and only need to calculate more measures related to the weights of items.

V. CONCLUSIONS AND FUTHER DEVELOPMENTS

In this article, the authors present an overview of common data structures used in mining frequent itemsets on the binary transaction database and a summary of some algorithms for mining frequent itemsets on transaction database weighted items in the direction of data structure analysis, search strategy on generate space, and method of calculating the related measures in researches over the past twenty years. The authors also give recommendations, and help researchers in data mining have enough knowledge when choosing an appropriate technical solution for the problem of mining frequent itemset on transaction database with weighted items.

From the summary and recommendations on some common frequent itemsets mining algorithms on transaction data with weighted items presented in Parts II and IV: In future, the authors will expand the research on some data structures and algorithms in order to mine frequent itemsets on large transaction database with weighted items, as well

as parallelize the proposed algorithm in order to quickly mine the frequent itemsets on multi-core processors, the distributed computing systems such as Hadoopp, Spark, etc.

REFERENCES

[1] R. Agrawal, T. Imilienski, A. Swami, "Mining association rules between sets of large databases", *Proc of the ACM SIGMOD Int Conf on Management of Data*, Washington, DC, 207-216, 1993.

[2] R. Agrawal, Srikant R., "Fast Algorithms for Mining Association Rules in Large Databases", *VLDB*, Chile, 487-499, 1994.

[3] J. Han, J. Pei, Y. Yin, "Mining Frequent Patterns without Candidate Generation", *SIGMOD Conf. 2000*, 1-12, 2000.

[4] J. Liu, Y. Pan, K. Wang, J. Han, "Mining frequent item sets by opportunistic projection", *KDD 2002*, 229-238, 2002.

[5] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal, "Discovering frequent closed itemsets for association rules", *Proceecings of the 5th Int Conf on Database Theory*, LNCS, Springer-Verlag, Jerusalem, Israel, 398-416, 1999.

[6] M.J. Zaki, C. J. Hsiao, "CHARM: An Efficient Algorithm for Closed Association Rule Mining", *Computer Science*, Rensselaer Polytechnic Institute 1-20, 1999.

[7] D. Lin, Z. M. Kedem, "Pincer-Search: A New Algorithm for Discovering the Maximum Frequent Itemset", *EDBT Conference Proceedings*, 105-110, 1998.

[8] R. J. Bayardo, "Efficiently mining long patterns from databases", *In Proc. The ACM SIGMOD Int. Conf. on Management of data*, 85-93, 1998.

[9] R. Agarwal, C. Aggarwal, V. V. V. Prasad, "Depth First Generation of Long Patterns", *In Proc. ACM SIGMOD*, 108-118, 2000.

[10] H. Phan, B. Le, "MAXLEN-FI: A fast algorithm for mining maximum length frequent itemsets", *DLU Journal of Science*, 8(2), 2018, 108-123.

[11] G. D. Ramkumar, S. Ranka, S. Tsur, "Weighted Association Rules: Model and Algorithm", *Proc. ACM SIGKDD*, 1998, 1-13.

[12] C. H. Cai, A. W. C. Fu, C. H. Cheng, W. W. Kwong, "Mining association rules with weighted items", *Proceedings. IDEAS'98. International Database Engineering and Applications Symposium (Cat. No.98EX156)*, Cardiff, UK, 68-77, 1998.

[13] F. Tao, F. Murtagh, M. M. Farid, "Weighted association rule mining using weighted support and significance framework",*SIGKDD'03*, 661-666, 2003.

[14] U. Yun, J. J. Leggett, "WFIM: Weighted Frequent Itemset Mining with a weight range and a minimum weight", *SDM 2005*, 636-640, 2005.

[15] R. Geng, X. Dong, X. Zhang, W. Xu, "Efficiently Mining Closed Frequent Patterns with Weight Constraint from Directed Graph Traversals Using Weighted FP-Tree Approach", *Computing Communication Control and Management 2008. CCCM'08. ISECS International Colloquium on*, 3, 399-403, 2008.

[16] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, Y. K. Lee, "Mining Weighted Frequent Patterns in Incremental Databases", *In: Ho TB., Zhou ZH. (eds) PRICAI 2008: Trends in Artificial Intelligence. PRICAI 2008. Lecture Notes in Computer Science*, vol 5351. Springer, Berlin, Heidelberg, 2008.

[17] B. Le, H. Nguyen, B. Vo, "Efficient Algorithms for Mining Frequent Weighted Itemsets from Weighted Items Databases", *RIVF 2010*, 1-6, .2010.

[18] B. S. Jeong, A. Farhan, "Efficient Dynamic Weighted Frequent Pattern Mining by using a Prefix-Tree", *The KIPS Transactions:PartD*, 17D, 253, 2010.

[19] R. Pears, Y. S. Koh, G. Dobbie, "EWGen: Automatic Generation of Item Weights for Weighted Association Rule Mining", *ADMA (1)*, 36-47, 2010.

[20] B. Wang, Z. Yuanpan, G. Feng Guo, "Mining weighted closed itemsets directly for association rules generation under weighted support framework", *IEEE 3rd Inter Conf on Communication Software and Networks*, 145-149, 2011.

[21] U. Yun, K. H. Ryu, "Approximate weighted frequent pattern mining with/without noisy environments", *Knowl. Based Syst*, 24(1), 2011, 73-82.

[22] Z. G. Huai, M. H. Huang, "A weighted frequent itemsets Incremental Updating Algorithm base on hash table", *2011 IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, 201-204, 2011.

[23] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, Y. K. Lee, H. J. Choi, "Single-pass incremental and interactive mining for weighted frequent patterns", *Expert Systems with Applications*, 39, 2012, 7976.

[24] U. Yun, H. Shin, K. H. Ryu, E. Yoon, "An efficient mining algorithm for maximal weighted frequent patterns in transactional databases", *Knowl. Based Syst*, 33, 2012, 53-64.

[25] B. Vo, F. Coenen, B. Le, "A new method for mining Frequent Weighted Itemsets based on WIT-trees", *Expert Systems with Applications*, 40(4), 2013, 1256-1264.

[26] B. Vo, N. Y. Tran, D. H. Ngo, "Mining Frequent Weighted Closed Itemsets", *Advanced Computational Methods for Knowledge Engineering 2013*, 379-390, 2013.

[27] G. C. Lan, T. P. Hong, H. Y. Lee, and C. W. Lin, "Mining Weighted Frequent Itemsets", *in Proceedings of the 30th workshop on Combinatorial Mathematics and Computation Theory (Alg'30)*, 85-89, 2013.

[28] U. Yun, K. H. Ryu, "Efficient mining of maximal correlated weight frequent patterns", *Intell. Data Anal*, 17(5), 2013, 917-939.

[29] A. Mohan, R. Visakh, "Weighted Frequent Pattern Mining using RDD, the Basic Spark Abstraction", *In Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '14)*. Association for Computing Machinery, New York, NY, USA, Article 81, 1-5, 2014.

[30] U. Yun, E. Yoon, "An Efficient Approach for Mining Weighted Approximate Closed Frequent Patterns Considering Noise Constraints", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 22(6), 2014, 879-912.

[31] D. H. Nguyen, B. Vo, T. H. M. Nguyen, T.-P. Hong, "An Improved Algorithm for Mining Frequent Weighted Itemsets", *IEEE International Conference on Systems, Man, and Cybernetics*, 2579-2584, 2015.

[32] G. C. Lan, T. P. Hong, H. Y. Lee, C. W. Lin, "Tightening upper bounds for mining weighted frequent itemsets", *Intell. Data Anal*, 19(2), 2015, 413-429.

[33] X. Wei, Z. Li, T. Zhou, H. Zhang, G. Yang, "IWFPM: Interested Weighted Frequent Pattern Mining with Multiple Supports", *Journal of Software*, 10, 2015, 9-19.

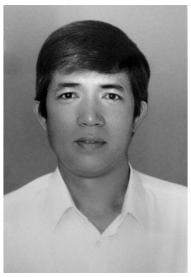
[34] G. Lee, U. Yun, H. Ryang, D. Kim, "Approximate Maximal Frequent Pattern Mining with Weight Conditions and Error Tolerance", *Int. J. Pattern Recognit. Artif. Intell*, 30(6), 2016, 1-42.

[35] H. Nguyen, B. Vo, M. Nguyen, W. Pedrycz, "An efficient algorithm for mining frequent weighted itemsets using interval word segments", *Appl. Intell*, 45(4), 2016, 1008-1020.

[36] U. Yun, G. Lee, "Incremental mining of weighted maximal frequent itemsets from dynamic databases", *Expert Syst. Appl*, 54, 2016, 304-327.

[37] Q. Qin, L. Tan, "An Efficient Mining Algorithm for Maximal Weighted Frequent Patterns Based on WIdT-Trees", *IDEAL*

2016, 596-605, 2016.
[38] U. Yun, G. Lee, K. M. Lee, "Efficient representative pattern mining based on weight and maximality conditions", *Expert Systems*, 33(5), 2016, 439-462.
[39] G. Lee, U. Yun, K. H. Ryu, "Mining Frequent Weighted Itemsets without Storing Transaction IDs and Generating Candidates", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 25(1), 2017, 111-144.
[40] J. C. W. Lin, W. Gan, P. F. Viger, H. C. Chao, T. P. Hong, "Efficiently mining frequent itemsets with weight and recency constraints", *Applied Intelligence*, 47, 2017, 769-792.
[41] B. Vo, "An Efficient Method for Mining Frequent Weighted Closed Itemsets from Weighted Item Transaction Databases", *J. Inf. Sci. Eng.*, 33(1), 2017, 199-216.
[42] H. Bui, B. Vo, H. Nguyen, T. A. N. Hoang, T. P. Hong, "A weighted N-list-based method for mining frequent weighted itemsets", *Expert Syst. Appl.*, 96, 2018, 388-405.
[43] R. U. Kiran, A. Kotni, P. K. Reddy, M. Toyoda, S. Bhall, M. Kitsuregawa, "Efficient discovery of weighted frequent itemsets in very large transactional databases: A re-visit", *Proc. IEEE Int. Conf. Big Data (Big Data)*, 723-732, 2018.
[44] X. Zhao, X. Zhang, P. Wang, S. Chen, Z. Sun, "A weighted frequent itemset mining algorithm for intelligent decision in smart systems", *in IEEE Access*, 6, 2018, 29271-29282.
[45] K. Klangwisan, K. Amphawan, "Efficient weighted-frequent-regular itemsets mining using interval word segments structure", *Knowledge and Smart Technology (KST) 2018 10th International Conference on*, 59-67, 2018.
[46] A. B. Cengiz, K. U. Birant, D. Birant, "Analysis of Pre-Weighted and Post-Weighted Association Rule Mining", *Innovations in Intelligent Systems and Applications Conference*, Izmir, Turkey, 1-5, 2019.
[47] U. Dewan, C. F. Ahmed, C. K. Leung, R. A. Rizvee, D. Deng, J. Souza, "An efficient approach for mining weighted frequent patterns with dynamic weights", *ICDM 2019*, 13-27, 2019.
[48] Z. Yue, J. Sun, R. Liu, "Mining Frequent Weighted Itemsets Using Extended N-List and Subsume", *International Conference on Robots & Intelligent System (ICRIS)*, 513-516, 2019.
[49] B. Vo, H. Bui, T. Vo, T. Le, "Mining top-rank-k frequent weighted itemsets using WN-list structures and an early pruning strategy", *Knowledge-Based Systems*, 201-202, 2020, 106064-106075.
[50] H. Bui, B. Vo, T. A. N. Hoang, U. Yun, "Mining frequent weighted closed itemsets using the WN-list structure and an early pruning strategy", *Appl Intell*, 51(3), 2021, 1439-1459.



Huan Phan is a Ph.D. Student at the Faculty of Mathematics and Computer Science, Vietnam National University Ho Chi Minh City - University of Science, Vietnam. His research interests: artificial intelligence, data mining and high performance computing.
Email: huanphan@hcmussh.edu.vn

Bac Le is a Professor at the Faculty of Information Technology Vietnam National University Ho Chi Minh City - University of Science, Vietnam. His research interests: artificial intelligence, soft computing, machine learning, data mining and data science.
Email: lhbac@fit.hcmuns.edu.vn