Noname manuscript No.
(will be inserted by the editor)

Linguistic Frequent Pattern Mining using a Compressed Structure

Jerry Chun-Wei Lin · Usman Ahmed · Gautam Srivastava · Jimmy Ming-Tai Wu · Tzung-Pei Hong · Youcef Djenouri

Received: date / Accepted: date

Abstract Traditional association-rule mining (ARM) considers only the frequency of items in a binary database, which provides insufficient knowledge for making efficient decisions and strategies. The mining of useful information from quantitative databases is not a trivial task compared to conventional algorithms in ARM. Fuzzy-set theory was invented to represent a more valuable form of knowledge for human reasoning, which can also be applied and utilized for quantitative databases. Many approaches have adopted fuzzy-set theory to

Jerry Chun-Wei Lin
Department of Computer Science, Electrical Engineering and Mathematical Sciences
Western Norway University of Applied Sciences, Bergen, Norway
E-mail: jerrylin@ieee.org

Usman Ahmed
Department of Computer Science, Electrical Engineering and Mathematical Sciences
Western Norway University of Applied Sciences, Bergen, Norway
E-mail: Usman.Ahmed@hvl.no

Gautam Srivastava
Department of Mathematics & Computer Science
Brandon University, Brandon, Canada
Research Centre for Interneural Computing
China Medical University, Taichung, Taiwan
E-mail: SRIVASTAVAG@brandonu.ca

Jimmy Ming-Tai Wu
College of Computer Science and Engineering
Shandong University of Science & Technology, Shandong, China
E-mail: wmt@wmt35.idv.tw

Tzung-Pei Hong
Department of Computer Science and Information Engineering
National University of Kaohsiung, Kaohsiung, Taiwan
E-mail: tphong@nuk.edu.tw

Youcef Djenouri
SINTEF Digital, Mathematics and Cybernetics, Oslo, Norway
E-mail: youcef.djenouri@sintef.no

transform the quantitative value into linguistic terms with its corresponding degree based on defined membership functions for the discovery of FFIs, also known as fuzzy frequent itemsets. Only linguistic terms with maximal scalar cardinality are considered in traditional fuzzy frequent itemset mining, but the uncertainty factor is not involved in past approaches. In this paper, an efficient fuzzy mining (EFM) algorithm is presented to quickly discover multiple FFIs from quantitative databases under type-2 fuzzy-set theory. A compressed fuzzy-list (CFL)-structure is developed to maintain complete information for rule generation. Two pruning techniques are developed for reducing the search space and speeding up the mining process. Several experiments are carried out to verify the efficiency and effectiveness of the designed approach in terms of runtime, the number of examined nodes, memory usage, and scalability under different minimum support thresholds and different linguistic terms used in the membership functions.

Keywords fuzzy-set theory · fuzzy data mining · fuzzy-list structure · pruning strategies

1 Introduction

Knowledge Discovery in Databases (KDD) [1,2,4,39,40,42] has been an important issue in many tasks since it can discover potential and implicit information from datasets. The first fundamental algorithm is known as Apriori [1], which is used to find associations of item(sets) in databases. Apriori uses the minimum support threshold to first identify the set of frequent itemsets (FIs), then apply the minimum confidence threshold to reveal the set of association rules (ARs) from the discovered FIs. An AR can thus be represented as $X \rightarrow Y$, where support through XY and confidence $X \rightarrow Y$ will be considered as no less than the pre-defined two thresholds. Here, both X and Y are the item(sets) represented in databases that are binary. Since Apriori is a level-wise approach, which needs higher computational costs to first generate the candidates then evaluates them level-by-level, an improved algorithm known as FP-growth [13] was implemented to improve mining efficiency by compressing relevant transactions into a tree structure (called FP-tree). Based on recursive FP-growth and compressed FP-tree structure, the k-itemsets can be recursively discovered.

Motivation and application: In a real-world application (complex environmental system, e.g. industrial sensor data), a wide variety of sensors are available that produce a massive amount of data. The produced dataset can make information mining and patterns analysis a more convenient task. The individual data sources have different uncertainty (data quantity) depending on the processing environment of different sensors. The information extraction, retrieval, and mining mostly used traditional mining based techniques to mine distinct patterns. The uncertainty factor assesses the reliability of patterns in terms of probability. Because of uncertainty associated with sensors resources (e.g. wireless sensor network, Wifi system, and RFID), it is not trivial to discover the meaningful and implicit information from databases. Also, analysis instead is based on the scanning of complete datasets (multiple scans) that requires a lot of computational resources for associated similarity and dissimilarity among data points. Since the size of sensor datasets grows with time, thus the computational cost to mine the required information increases as well with time. The mining procedures associated similar issues with fuzzy type systems (e.g., type-1 fuzzy sets) is used for solving uncertainty with probability interpretations of a point [45]. The fuzzy type-1 membership function handles the values within the range $[0, 1]$ for uncertainty measures. However, fuzzy type-1 still has interpretability issues as a membership function remains uncertain under different conditions [9]. The interpretability issues are resolved with the usage of a fuzzy type-2 membership function. The fuzzy type-2 second membership value makes it computationally less expensive throughout the domain. The fuzzy type-2 membership function has the uncertainty factor to produce an interval for the fuzzy degrees (upper and lower values) by the utilization of the pre-defined membership functions. The utilization of the fuzzy type-2 membership function with a comprehensive list structure helps to encompass all the data-points and accommodate different data generated by sensors. It also helps to incorporate missing or uncertain points if results suffer from any type of hardware failure. It can encompass the missing information within a particular proximity. Thus, the uncertain factor can be involved and considered. Furthermore, with the help of a compressed data structure, a less number of scans is then required to handle the mining progress in big datasets, including the uncertainty factor associated with the data and its exponential growth.

For most works regarding ARM, the focus is mostly centered around the mining of FIs or ARs from binary databases, which only considers whether an item(set) appears in the databases. The other important factors such as interestingness, weight, importantness, and quantity are not considered as major factors in ARM. Thus, the discovered information such as FIs or ARs can thus be used for making inefficient or wrong decisions since the discovered knowledge may be insufficient and incomplete. In real-life domains and applications, an item can be purchased with several amounts in shopping behaviours, for instance, as an example, suppose a patron buys **five** bottles of beer or **two** cartons of milk. It is thus not a trivial task to discover knowledge and information from the quantitative databases. Fuzzy-set theory [10,23,45] was thus designed and used in many intelligent systems such as in engineering fields, manufacturing, and/or medical diagnosis since the represented knowledge based on fuzzy-sets is more interpretable for human reasoning. Furthermore, it can be used for the conversion of quantitative values of items to linguistic terms in nature with corresponding degrees, which is easier for managers and retails to make efficient decisions. Hong *et al.* [12] designed an algorithm that uses the Apriori-like approach to level-wisely discover the set of fuzzy association rules (FARs). It considers terms that are linguistic with cardinality (maximal scalar) of items able to clearly show its linguistic variable. Based on the maximal scalar cardinality, the computational cost can be reduced, and the # of derived linguistic terms remains the same number as the # of original database items. To speed up computations, Lin *et al.* next implemented a fuzzy frequent pattern tree (FFPT) [21], compressed FFPT (CFFPT) [22], and an upper-bound FFP tree (UBFFPT) [24] which is used to improve the performance for mining of FFIs. Many methods were respectively developed to mine FFIs based on different structures and pruning strategies to reduce computational cost. However, the above approaches only consider one linguistic term with the maximal scalar cardinality of an item, thus for decision-making purposes, the information which is discovered may only be partial. Several algorithms considered multiple fuzzy frequent itemsets (MFFIs) [15,16,25,26] to derive more complete and sufficient knowledge. Therefore, suppose the fuzzy value of a term that is linguistic of an item is great than the support threshold considered as a minimum, it will be treated as a frequent itemset. Based on this mechanism, more complete rules can be mined, and useful decisions can thus be produced.

The above methods mostly consider the fuzzy set theory (type-1) to discover required information and knowledge, i.e., ARs or FIs. However, the algorithms use the conventional type-1 fuzzy-sets currently as well as a linguistic term with a discrete value. Mendel then designed type-2 fuzzy-set theory [34] by involving the uncertain factor to mine required information for decision-making. Chen *et al.* [7] integrated the type-2 fuzzy-sets model and considered the pattern mining problem to handle quantitative databases based on the level-wise approach. However, this approach still holds the single-linguistic term of each item for knowledge presentation, thus derived information may still be incomplete. Lin *et al.* [28] was able to create a list method for efficiently mining type-2 fuzzy frequent patterns, which can increase mining performance when the directly side-by-side comparison is shown with the level-wise approach. It does not, however, have successful pruning methods to prune the search space for pattern discovery. The authors, however, still explore many unpromising candidates.

In this paper, we present a compressed fuzzy-list (CFL)-structure to keep more information for subsequent mining processes. Two effective pruning strategies and an efficient mining (EFM) algorithm have been developed to mine the multiple fuzzy frequent patterns (MFFPs). Major contributions of this paper are summarized below:

1. An efficient fuzzy mining (EFM) method is presented to discover multiple fuzzy frequent patterns (MFFPs) efficiently considering the uncertainty based on fuzzy-sets (type-2).
2. A compressed (CFL)-structure (fuzzy) is shown to keep the condensed upper-bound value on the potential candidates for subsequent mining processes.
3. Two effective CFL-based pruning strategies are then built, to deduct the size of the search space, thus dramatically decreases the computational cost.

4. Experiments are conducted to show that the designed approach outperforms the level-wise-like and conventional list-based approaches in terms of runtime and number of examined candidates.

The remainder of this paper is structured in the following sequence. In Section 2, the literature is briefly discussed and reviewed. Through work in Section 3, the preliminary and problem statement of FFPM (fuzzy) are given. Section 4 describes the structure, algorithm, and pruning strategies that have been developed. Experiments in Section 5 are carried out and presented. The conclusion and future work will finally be drawn in Section 6.

2 Literature Review

As the rapid growth of information techniques [32,33], it is an interesting topic to reveal the relationship of the itemsets in the databases. ARM, known in long-hand as Association-Rule Mining [1, 2, 4] is a basic methodology used in knowledge discovery, which shows the relationships among itemsets in binary databases. The first algorithm is known as Apriori [2], which uses a "level-wise approach" to discover numerous association rules (ARs). It uses the minimum support threshold to first mine the set of frequent itemsets (FIs), then applies the minimum confidence threshold to explore the ARs from the discovered FIs. This approach is continued by a level-wise approach. Thus, the computational cost is very high to produce ARs. To solve the limitation of Apriori, FP-growth [13] was presented to speed up mining performance. It uses the FP-tree structure to keep the frequent 1-itemsets then mines the set of FIs from the conditional FP-tree structure level-by-level. Several extensions of frequent itemsets mining (FIM) are then further studied and developed in many different applications and domains [20,29,30]. Most of the methodologies focus on mining the required information from binary databases. In realistic situations, an item may, however, be purchased with several quantities in a transaction [8, 31, 42]. It is thus a non-trivial task to retrieve the information from the quantitative databases since DC, short for downward closure, which is required for maintenance of ensuring the correctness and completeness of the discovered knowledge.

In the last 20 years, fuzzy-set theory [10, 45] is effective in many areas since it is interpretable for human reasoning. Fuzzy-set theory is an extension of the conventional crisp set by identifying linguistic membership functions and their corresponding membership degrees (range from 0 to 1) based on the membership functions themselves. The fuzzy-set theory considers quantifying and reasoning using linguistic terms with the corresponding membership degrees (fuzzy values). Several algorithms (both fuzzy and/or mining) have been shown to produce interesting rules which have been extensively discussed and developed. Srikant *et al.* [36] introduced the approach for defining ARs by partitioning and transforming the problem with a binary database. Au *et al.* [3] designed F-APACS which is used to mine ARs that are fuzzy (FARs) by using linguistic terms to find both exceptions as well as regularities, which can be more meaningful for human experts to understand the mined knowledge. Kuok *et al.* [18] developed an algorithm to process the quantitative attributes and showed that the fuzzy-sets have a stronger capability to deal with values when compared to other methods. Hong *et al.* [12] implemented a fuzzy mining algorithm that mines rules based on the "generate-and-test" approach for handling quantitative databases then proposed a GDF approach [15] to efficiently discover the set of multiple fuzzy frequent itemsets (MFFIs). The GDF uses the gradual concept to mine the MFFIs that also reduces the size of the processed database gradually; the computational cost can thus be reduced since some unpromising linguistic terms can also be deducted together in the mining progress. Chen *et al.* [6] developed a novel model that fused other models, which is used to improve mining procedures. The rules are multi-level as well as fuzzy built on cumulative information. Watanabe *et al.* [41] has established the redundancy equivalence and theorems for FARs. The Apriori-like method was applied to use the redundancy equivalence of items (fuzzy) through the use of the principles of redundancy in the discovery of FARs. Mishra *et al.* [35] also implemented a frequent pattern mining method for handling a fuzzified gene expression and showed that the vertical fuzzy dataset format could produce more fuzzy FIs than the original one. Gupta and Muhuri used Tsukamoto's inference method to analyze student academic performance [9]. The method used multi-objective linguistic optimization problems (MOLOPs) based on the 2-tuple fuzzy linguistic approach for monotonic and non-monotonic functions. The authors show the proposed method with student performance evaluation. Shukla and Muhuri also addressed the uncertainty factor in big datasets using fuzzy type-2 sets [37]. The proposed method is used to handle the veracity issues in the big dataset. The methods use the concept of the footprint of uncertainty in interval type-2 fuzzy sets [37]. The method is then evaluated regarding consistency and efficacy with different aspects, which handles veracity issues and is efficient in reducing instances. Several algorithms based on the fuzzy-set theory for mining the required information in different applications and domains were then studied and developed in progress [5, 11, 19, 27, 38, 43].

To speed up the **generate and test** methodology for mining the FFIs, Lin *et al.* then developed the fuzzy frequent pattern tree (FFP)-tree algorithm [21] to compress the fuzzy 1-itemsets into a tree structure for later mining process. The transformed terms (fuzzy linguistic 1-itemsets) with their values are ordered (ascending) for every transaction. However, the given approach has produced a loose tree structure. Thus a compressed CFFP-tree algorithm [22] was proposed in an attempt to reduce the size of all the nodes in the tree. An array is used to keep more information about each node. Thus, the fuzzy values are preserved consequently. This process can greatly reduce the computational cost of mining performance. However, this approach still needs extra memory usage for the attached array. Consequently, it sometimes has the dreaded memory leakage problem. As a solution, the upper-bounded FFP tree (UBFFP)-tree algorithm [24] was created to ensure a higher condenses structure of the tree, thus reducing the memory leakage problem for handling big datasets.

The above works only work on the type-1 fuzzy-set theory, where uncertainty is not considered as a factor. The functions for membership of set theory (fuzzy type-1) are entirely sharp, which is inadequate in realistic applications to manage uncertainty models. For instance, sensed information from various sensors could be affected by environmental factors. (i.e., snow, storms, or rain). To better present discovered knowledge with uncertainty, set theory (type-2 fuzzy) [14, 17, 34] was invented and established concurrently. To incorporate type-2 fuzzy-sets with pattern mining, Chen *et al.* [7] first developed a conventional level-wise (or Apriori-like) approach to mine fuzzy type-2 frequent patterns level-wisely. This approach requires to generate many unpromising candidates with highly computational cost, which is not efficient for any sort of mining tasks. Moreover, it uses the maximal scalar cardinality approach to retrieve only a term (single linguistic) of a given item, which for all intents and purposes should create a lack of actual knowledge for decision-making. Lin *et al.* [28] then gave a list-based approach to maintain complete information for subsequent mining processes. However, without efficient pruning strategies and the tighter upper-bound value on unpromising patterns, this approach still has to examine many candidates for deriving actual fuzzy frequent patterns.

3 Preliminaries and Problem Statement

To better understand the paper's notation that is used, a notion table is given in Table 1.

Table 1: A notation table

Sybmol	Description
D	the database in which $D = \{T_1, T_2, \ldots, T_n\}$.
I	the items in the database in which $I = \{i_1, i_2, \ldots, i_m\}$.
v_{i_T}	the quantity of the item i in transaction T.
X	the set of the items in which $X = \{i_1, i_2, \ldots, i_k\}$.
δ	the minimum support threshold.
μ	the defined membership function.
f_{i_T}	the fuzzy linguistic terms of item i in transaction T.
$fv_{i_T l}^{lower}$	the lower membership degree of v_{i_T} for an item i in the l-th fuzzy terms.
$fv_{i_T l}^{upper}$	the upper membership degree of v_{i_T} for an item i in the l-th fuzzy terms.
R_{il}	the l-th fuzzy term of i in μ.
$fv_{i_T l}^{c}$	the degree of fuzzy term R_{il}.
$Sup(R_{jl})$	the scalar cardinality of R_{il}.
$fv(X)$	the fuzzy membership value of X in T.
$mrfv(X,T)$	the maximum remaining fuzzy value of X in T.
$rmrfv(X,T)$	the relative maximum remaining fuzzy value of X in T.
$Sup(X)$	the sum up value of *mrfv* of X.
$rSup(X)$	the sum up value of *rmrfv* of X.

We can assume I is given as a set finite in nature with m distinct items in the database D. To better present the following content, i is then used to represent each item in the database D. The database with quantitative values of the items is considered as D, in which D has n transactions. Each item i in T has its purchase amount, which is denoted as v_{i_T}. A k-itemset is denoted as X, in which each $X \subseteq I$. Without the quantitative value of i in a transaction T, X must appear in any of the combinations of i in T. A membership functions used in type-2 fuzzy-set theory is denoted as μ. A threshold δ is used as the minimum support to verify whether an itemset is considered as the fuzzy frequent pattern. A simple example is illustrated in Table 2, which consists of ten transactions and six distinct items, denoted from a to f.

Table 2: An illustrated quantitative database.

TID	Items with the purchase amounts
T_1	a:5, c:4, e:1
T_2	a:3, e:1
T_3	a:1, e:2, f:2
T_4	b:2, c:1, e:3
T_5	a:4, b:5, c:5, d:3, e:3
T_6	b:4, d:1, e:4
T_7	c:4, e:2
T_8	b:4, e:4, f:3
T_9	b:3, c:4, e:2, f:1
T_{10}	e:5, f:5

Suppose that the minimum support threshold in Table 2 is set as δ (= 20%), and the type-2 fuzzy-sets used in the example are illustrated in Fig. 1. Here, 3 terms called $L - Low$, $M - Middle$, and $H - High$ which are given as part of μ. We address here that a user can specify the number of terms based on a variety of different requirements.

Definition 1 The v_{i_T} is represented as the quantitative value of i, which shows the quantitative of the item (linguistic variable) i in a transaction T.

For instance, the quantitative values of the items $(a), (c)$, and (e) in transaction 1 respectively are and $v_{a_{T_1}}(= 5)$, $v_{c_{T_1}}(= 4)$, and $v_{e_{T_1}}(= 1)$.

Definition 2 The f_{i_T} is considered as the set of fuzzy linguistic terms with their membership degrees (fuzzy values) that was transformed from the quantitative value v_{i_T} of the linguistic variable i by μ as:

$$f_{i_T} = \mu_i(v_{i_T})(= \frac{(fv_{i_T1}^{lower}, fv_{i_T1}^{upper})}{R_{i1}} + \frac{(fv_{i_T2}^{lower}, fv_{i_T2}^{upper})}{R_{i2}} + \cdots + \frac{(fv_{i_Th}^{lower}, fv_{i_Th}^{upper})}{R_{ih}}), \tag{1}$$

in which h represents the number of fuzzy terms of i transformed by μ, R_{il} shows the l-th fuzzy terms of i, $v_{i_Tl}^{lower}$ indicates the lower membership degree (fuzzy value) of v_{i_T} for i in the l-th fuzzy terms R_{il}, $fv_{i_Tl}^{upper}$ states the upper membership degree (fuzzy value) of v_{i_T} for i in the l-th fuzzy terms R_{il}, $fv_{i_Tl}^{lower} \leq fv_{i_Tl}^{upper}$, and $fv_{i_Tl}^{lower}, fv_{i_Tl}^{upper} \subseteq [0, 1]$.

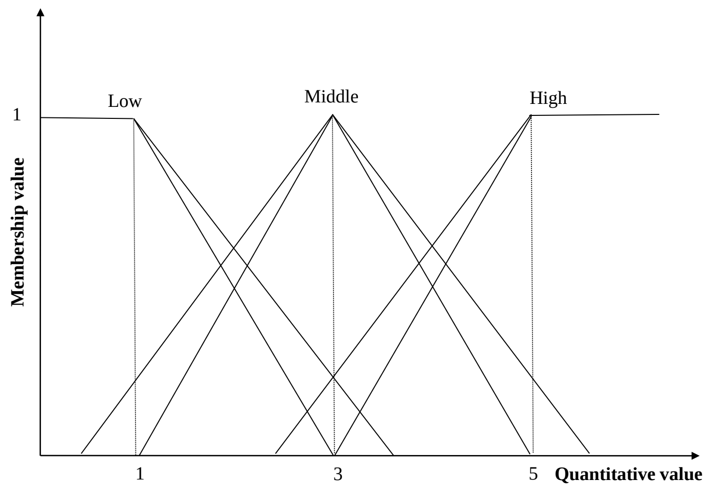

Fig. 1: An illustrated membership functions with (L), (M), and (H) linguistic terms.

Note that the $fv_{i_T l}^{lower}$ and $fv_{i_T l}^{upper}$ are respectively two membership degrees for the fuzzy term R_{il}. For instance, the item (c) with its quantitative value 4 in T_1 is transformed by the membership functions in Fig. 1 as $(\frac{(0.5,0.63)}{c.M} + \frac{(0.5,0.63)}{c.H})$, where only two fuzzy terms $(c.M)$ and $(c.H)$ are considered here; $(c.M)$ is the fuzzy term for the membership degree as (0.5,0.63). The lower value is 0.5 and upper value is 0.63 for $(c.M)$; $(c.H)$ is the fuzzy term for the membership degree as (0.5,0.63). The lower value is 0.5 and upper value is 0.63 for $(c.H)$. We can also observe that the lower membership degree (0.5) is less than the upper membership degree (0.63) such that $0.5 < 0.63$.

Definition 3 The i is an attribute (item) in the database such that $i \in I$, which is also treated as the linguistic variable, and its value is the set of fuzzy terms represented as the natural language such that $R_{i1}, R_{i2}, \ldots, R_{ih}$. These fuzzy terms can be transformed by the pre-defined μ (membership functions).

For instance, six linguistic variables (attributes) such as (a), (b), (c), (d), (e), and (f) are denoted in Table 2 and three linguistic terms of L, M and H are defined in Fig. 1. In this membership function, suppose an item is set as X, and if the quantitative value is set as 1, the it is then converted as $(\frac{(1,1)}{X.L}) + \frac{(0,0.25)}{X.M}$; if the quantitative value is set as 2, it is then converted as $\frac{(0.5,0.63)}{X.L} + \frac{(0.5,0.63)}{X.M}$; if the quantitative value is set as 3, it is then converted as $\frac{(0,0.25)}{X.L} + \frac{(1,1)}{X.M} + \frac{(0,0.25)}{X.H}$; if the quantitative value is set as 4, it is then converted as $\frac{(0.5,0.63)}{X.M} + \frac{(0.5,0.63)}{X.H}$; and if the quantitative value is set as 5, it is then converted as $\frac{(0,0.25)}{X.M} + \frac{(1,1)}{X.H}$. Note that the membership functions can be defined by users' preference and the specific domains and applications, it is appropriate to present it by a figure.

For the given example in Table 2, each transaction in the database is then transformed by the membership functions of Fig. 1. The final results after transformation are shown in Table 3.

Table 3: Table 2 as a transformed database

TID	Linguistic fuzzy transformed terms
T_1	$\frac{(0,0.25)}{a.M}+\frac{(1,1)}{a.H}, \frac{(0.5,0.63)}{c.M}+\frac{(0.5,0.63)}{c.H}, \frac{(1,1)}{e.L}+\frac{(0,0.25)}{e.M}$
T_2	$\frac{(0,0.25)}{a.L}+\frac{(1,1)}{a.M}+\frac{(0,0.25)}{a.H}, \frac{(1,1)}{e.L}+\frac{(0,0.25)}{e.M}$
T_3	$\frac{(1,1)}{a.L}+\frac{(0,0.25)}{a.M}, \frac{(0.5,0.63)}{e.L}+\frac{(0.5,0.63)}{e.M}, \frac{(0.5,0.63)}{f.L}+\frac{(0.5,0.63)}{f.M}$
T_4	$\frac{(0.5,0.63)}{b.L}+\frac{(0.5,0.63)}{b.M}, \frac{(1,1)}{c.L}+\frac{(0,0.25)}{c.M}, \frac{(0,0.25)}{e.L}+\frac{(1,1)}{e.M}+\frac{(0,0.25)}{e.H}$
T_5	$\frac{(0.5,0.63)}{a.M}+\frac{(0.5,0.63)}{a.H}, \frac{(0,0.25)}{b.M}+\frac{(1,1)}{b.H}, \frac{(0,0.25)}{c.M}+\frac{(1,1)}{c.H}, \frac{(0,0.25)}{d.L}+\frac{(1,1)}{d.M}+\frac{(0,0.25)}{d.H}, \frac{(0,0.25)}{e.L}+\frac{(1,1)}{e.M}+\frac{(0,0.25)}{e.H}$
T_6	$\frac{(0.5,0.63)}{b.M}+\frac{(0.5,0.63)}{b.H}, \frac{(1,1)}{d.L}+\frac{(0,0.25)}{d.M}, \frac{(0.5,0.63)}{e.M}+\frac{(0.5,0.63)}{e.H}$
T_7	$\frac{(0.5,0.63)}{c.M}+\frac{(0.5,0.63)}{c.H}, \frac{(0.5,0.63)}{e.L}+\frac{(0.5,0.63)}{e.M}$
T_8	$\frac{(0.5,0.63)}{b.M}+\frac{(0.5,0.63)}{b.H}, \frac{(0.5,0.63)}{e.M}+\frac{(0.5,0.63)}{e.H}, \frac{(0,0.25)}{f.L}+\frac{(1,1)}{f.M}+\frac{(0,0.25)}{f.H}$
T_9	$\frac{(0,0.25)}{b.L}+\frac{(1,1)}{b.M}+\frac{(0,0.25)}{b.H}, \frac{(0.5,0.63)}{c.M}+\frac{(0.5,0.63)}{c.H}, \frac{(0.5,0.63)}{e.L}+\frac{(0.5,0.63)}{e.M}, \frac{(1,1)}{f.L}+\frac{(0,0.25)}{f.M}$
T_{10}	$\frac{(0,0.25)}{e.M}+\frac{(1,1)}{e.H}, \frac{(0,0.25)}{f.M}+\frac{(1,1)}{f.H}$

Lin *et al.* [28] developed a list-based structure to mine multiple fuzzy frequent patterns based on type-2 fuzzy sets. However, this methodology does not provide efficient pruning strategies to reduce the size of the search space. Consequently, many unpromising candidates are still examined. Moreover, the upper-bound values on the candidates are over-estimated. Thus, the problem statement of this paper is described next.

Problem Statement: The problem statement is formally defined as follows:
Input: The quantitative database D, the type-2 membership functions μ, and the minimum support threshold δ.
Output: The set of the discovered fuzzy frequent itemsets.
Objectives: Design a compressed data structure to keep the complete information from D; several pruning strategies to reduce the search space and the computational cost in the mining progress.

4 Proposed efficient fuzzy mining (EFM) algorithm

The purchase amount is considered as the quantitative value that will be transformed into the linguistic terms (variables) with the relevant fuzzy values (degrees for the linguistic terms) based on the pre-defined membership functions. Different linguistic terms will be pre-defined based on users' preferences in the membership functions. For instance, the database is shown in Table 2 was then taken through a transformation process using the membership functions of the type-2 fuzzy-set shown in Fig. 1. After that, results are stated in Table 3. Since it is not a trivial task to elaborate the interval fuzzy value in the mining progress, the centroid type-reduction method [7] is then applied to reduce the complexity for mining MFFPs of the interval values. The definition is stated as follows.

Definition 4 We can define the degree of membership of a given linguistic term R_{il} in a database (transformed) D' is clearly noted as $fv^c_{i_T l}$, and defines as:

$$fv^c_{i_T l} = \frac{fv^{lower}_{i_T l} + fv^{upper}_{i_T l}}{2}. \tag{2}$$

For example in transaction T_1 of the very first given Table 2, item (c) with its own quantity 4 which then goes through a transformation process as $\frac{(0.5,0.63)}{c.M} + \frac{(0.5,0.63)}{c.H}$. After that, the interval (0.5, 0.63) including $c.M$ and $c.H$ goes through a reduction process as $\frac{0.5+0.63}{2} = 0.56$ using centroid type reduction methodology. The linguistic term's value that is fuzzy as given in Table 3 is further processed which leads to the results as shown in Table 4.

Table 4: A revised database.

TID	Transformed linguistic terms
T_1	$\frac{0.13}{a.M} + \frac{1}{a.H}, \frac{0.56}{c.M} + \frac{0.56}{c.H}, \frac{1}{e.L} + \frac{0.13}{e.M}$
T_2	$\frac{0.13}{a.L} + \frac{1}{a.M} + \frac{0.13}{a.H}, \frac{1}{e.L} + \frac{0.13}{e.M}$
T_3	$\frac{1}{a.L} + \frac{0.13}{a.M}, \frac{0.56}{e.L} + \frac{0.56}{e.M}, \frac{0.56}{f.L} + \frac{0.56}{f.M}$
T_4	$\frac{0.56}{b.L} + \frac{0.56}{b.M}, \frac{1}{c.L} + \frac{0.13}{c.M}, \frac{0.13}{e.L} + \frac{1}{e.M} + \frac{0.13}{e.H}$
T_5	$\frac{0.56}{a.M} + \frac{0.56}{a.H}, \frac{0.13}{b.M} + \frac{1}{b.H}, \frac{0.13}{c.M} + \frac{1}{c.H}, \frac{0.13}{d.L} + \frac{1}{d.M} + \frac{0.13}{d.H}, \frac{0.13}{e.L} + \frac{1}{e.M} + \frac{0.13}{e.H}$
T_6	$\frac{0.56}{b.M} + \frac{0.56}{b.H}, \frac{1}{d.L} + \frac{0.13}{d.M}, \frac{0.56}{e.M} + \frac{0.56}{e.H}$
T_7	$\frac{0.56}{c.M} + \frac{0.56}{c.H}, \frac{0.56}{e.L} + \frac{0.56}{e.M}$
T_8	$\frac{0.56}{b.M} + \frac{0.56}{b.H}, \frac{0.56}{e.M} + \frac{0.56}{e.H}, \frac{0.13}{f.L} + \frac{1}{f.M} + \frac{0.13}{f.H}$
T_9	$\frac{0.42}{b.L} + \frac{0.71}{b.M}, \frac{0.56}{c.M} + \frac{0.56}{c.H}, \frac{0.56}{e.L} + \frac{0.56}{e.M}, \frac{(1}{f.L} + \frac{0.13}{f.M}$
T_{10}	$\frac{0.13}{e.M} + \frac{1}{e.H}, \frac{0.13}{f.M} + \frac{1}{f.H}$

To evaluate whether a pattern is an MFFP, the cardinality which is scalar for every term (linguistic) is next summed up for evaluation. We give useful definitions next.

Definition 5 The scalar cardinality of each linguistic term is the summed up value of the transformed membership degrees and can be represented as the support value of a linguistic term as:

$$Sup(R_{jl}) = \sum_{R_{jl} \subseteq T_r \wedge T_r \in D'} fv^c_{iql}, \tag{3}$$

To discover the complete information of MFFPs, the multiple linguistic terms of an item(set) is considered in the derived knowledge. The strategy called *MultiTerm* is then adopted here to keep the complete information for later mining progress of F2FPs, which is described next.

Strategy 1 (Multiple terms with scalar cardinality, MultiTerm) To mine more and complete information, each linguistic term R_{in} of an item i, whose scalar cardinality (Sup) is no less the predefined minimum support count ($minSup \times |D|$) is considered to be represented of the item. Thus, each linguistic may have at least one represented fuzzy term with its membership degree (fuzzy value).

For example in Table 4, the minimum support threshold is set as 20%. Thus, the minimum support value is calculated as $0.2 \times 10 (= 2)$. For instance, the $Sup(c.H)$ $(= 2.68 > 2)$, $Sup(e.L)(= 3.94 > 2)$, $Sup(e.M)(= 5.19 > 2)$, and $Sup(e.H)(= 2.94 > 2)$ satisfy the condition and are considered as MFFPs. Based on this strategy, the multiple fuzzy frequent itemsets can thus be discovered and used to provide more complete information for decision-making.

To maintain the downward closure property for building the compressed fuzzy-list (CFL)-structure, the linguistic terms in the transactions are sorted in order (ascending) by *ASCorder* strategy, which is described next.

Strategy 2 (Sort in ascending order, ASCorder) Each linguistic term of transactions in the transformed database D' is then sorted in ascending order of their support value, and denoted as $\prec$ which can be used for later processing of CFL-structure construction phase.

For example, the terms that are remaining of the entire transaction set as shown in Table 4 next go through a sorting procedure (ascending) of their given support values. The revised and sorted transactions are indicated in Table 5.

Table 5: The sorted database.

TID	**Linguistic terms**
T_1	$\frac{0.56}{c.H}, \frac{1}{e.L}, \frac{0.13}{e.M}$
T_2	$\frac{1}{e.L}, \frac{0.13}{e.M}$
T_3	$\frac{0.56}{e.L}, \frac{0.56}{e.M}$
T_4	$\frac{0.13}{b.M}, \frac{0.13}{e.H}, \frac{0.13}{e.L}, \frac{1}{e.M}$
T_5	$\frac{0.13}{b.M}, \frac{0.13}{e.H}, \frac{1}{c.H}, \frac{0.13}{e.L}, \frac{1}{e.M}$
T_6	$\frac{0.71}{b.M}, \frac{0.56}{e.H}, \frac{0.56}{e.M}$
T_7	$\frac{0.56}{c.H}, \frac{0.56}{e.L}, \frac{0.56}{e.M}$
T_8	$\frac{0.71}{b.M}, \frac{0.56}{e.H}, \frac{0.56}{e.M}$
T_9	$\frac{0.71}{b.M}, \frac{0.56}{c.H}, \frac{0.56}{e.L}, \frac{0.56}{e.M}$
T_{10}	$\frac{1}{e.H}, \frac{0.13}{e.M}$

After the original database is revised and sorted, the algorithm is processed to construct the CFL-structure. Each remaining 1-itemset is used to construct its relevant CFL-structure for maintaining the complete information. Properties of the CFL-structure are given next.

Definition 6 Assume that X is considered as the set of the linguistic terms and T is set as a transaction such that $X \subseteq T$. Thus, the remaining set for all linguistic terms in T after X is denoted as T/X.

For instance in Table 5, $T_1/(c.H) = (e.L,\, e.M)$ and $T_1/(e.L) = (e.M)$.

Definition 7 The maximum remaining fuzzy value of X in T, denoted as $mrfv(X, T)$, is the maximum fuzzy membership value of all terms in T/X as $mrfv(X, T) = max(fv(i, T/X))$.

Definition 8 The *relative maximum remaining fuzzy value* of X in T, denoted as $rmrfv(X, T)$, is the minimum fuzzy membership value between $mrfv(X, T)$ and $fv(X, T)$.

The definition of the developed CFL-structure is then described in Definition 9.

Definition 9 Each element in the CFL-structure of X has three attributes (ordered) as: tid, fv, and $rmrfv$.

- *tid* shows that the term X is in a transaction T.
- *fv* shows the fuzzy membership value of X in a transaction T.
- *rmrfv* shows the relative maximum remaining fuzzy membership value after X in a transaction T, which is the minimum value between $mrfv(X, T)$ and $fv(X, T)$.

Here, Sup is defined as the sum up value of fv in the CFL-list structure, and $rSup$ is the sum up value of $rmrfv$ in the CFL-list structure. From Definition 9, the new developed CFL-structure is given in Fig. 2. For instance as we show clearly through Fig. 2, the fuzzy term $(b.M)$ appears in transactions T_4, T_5, T_6, T_8, and T_9, and its elements are (4, 0.13, 0.13), (5, 0.13, 0.13), (6, 0.71, 0.56), (8, 0.71, 0.56) and (9, 0.71, 0.56), respectively. The Sup and $rSup$ are 0.239 and 0.194. In this example, the Sup is greater than the $minSup$ $(= 0.2)$ that means the $(b.M)$ is considered as the MFFP. However, since its $rSup$ is less than 0.2, it is not necessary to explore the extensions of $(b.M)$; the size of the search space can thus be greatly deducted. The construction algorithm of the CFL-structure is then stated in Algorithm 1.

Algorithm 1: Construction of the 1-pattern in the CFL-structure.

```
Input: D', a revised and sorted dataset.
Output: the CFLs-structures and large 1-patterns L'.
for each linguistic term t_jn of item j do
    if Sup(t_jn) ≥ minSup then
        put t_jn into L', and keep L' as Sup-ascending order;
for each linguistic term t_jn of L' in each T of D' do
    add element (tid, fv of t_jn in T, rmrfv of t_jn in T) to t_jn-CFL-structure;
    CFLs = CFLs ⋃ t_jn-CFL-structure;
return L', constructed CFLs ;
```

After CFL-structures are generated, a pruning strategy will be taken to reduce the space searching, which uses the $Supt$ and $rSup$ of such a list X

b.M		
4	0.13	0.13
5	0.13	0.13
6	0.71	0.56
8	0.71	0.56
9	0.71	0.56

e.H		
4	0.13	0
5	0.13	0.13
6	0.56	0
8	0.56	0
10	1	0

tids *fv* *rmrfv*

c.H		
1	0.56	0.56
5	1	1
7	0.56	0.56
9	0.56	0.56

e.L		
1	1	0
2	1	0
3	0.56	0
4	0.13	0
5	0.13	0
7	0.56	0
9	0.56	0

e.M		
1	0.13	0
2	0.13	0
3	0.56	0
4	1	0
5	1	0
6	0.56	0
7	0.56	0
8	0.56	0
9	0.56	0
10	0.13	0

Fig. 2: A built CFL-structure.

to decide whether to search the extension of X. The strategy is described as Lemma 1.

Definition 10 A termset is considered as the combinations of the linguistic terms (variables), forming as k-itemsets ($k \geq 1$) in the database.

Lemma 1 For an termset X, if $Sup(X)$ or $rSup(X)$ is less than the minimum support threshold, then any supersets (extension) of X is not multiple fuzzy frequent pattern and should be pruned.

From the given example, the search space for mining the required MFFPs is based on the enumeration tree, which is shown in Fig. 3.

To perform and generate the k-itemsets($k \geq 2$), the terms of P_x and P_y are used to generate the CFL-structure, forming as P_{xy}. The fuzzy terms are first examined to determine whether the valid $P_{xy}.CFL$ is generated. If P_x and P_y appear in the same transactions (TIDs), the simple join operation is then performed to calculate the fv of each transaction T. Furthermore, the minimum operation is also adopted to find the remaining $rmrfv$ of the P_{xy} in T. This process is then described next.

- $E_{xy}.tid$=$E_x.tid$ (or $E_y.tid$).
- $E_{xy}.fv$=$min(E_x.tid$, $E_y.tid)$.
- $E_{xy}.rmrfv$ = $min(E_x.rmrfv$, $E_y.rmrfv)$.

Here, we can note that if the sum of fv is no larger than the predefined minimum support count, it is not considered as the MFFP and the supersets will be discarded and ignored, directly without any further exploration. This progress is then executed recursively until no candidates can be generated. The details are stated in Algorithm 2.

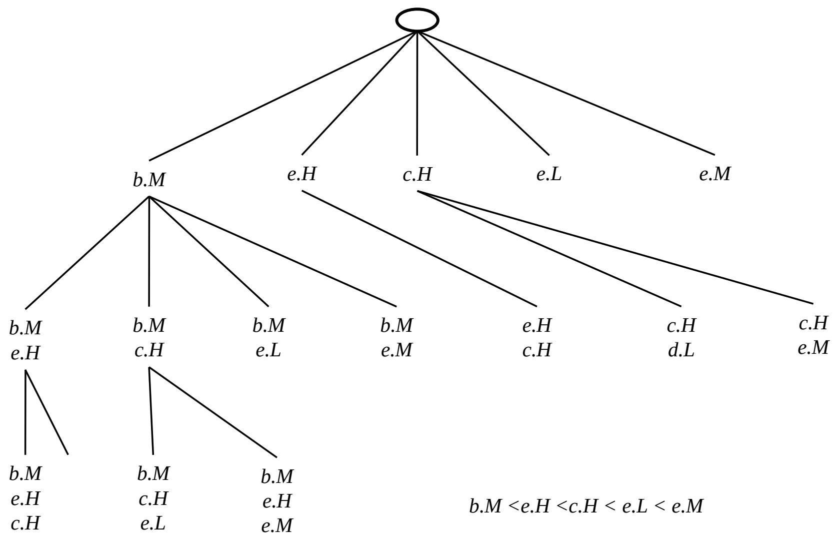

Fig. 3: The size of search space in the running example.

Algorithm 2: Construct($P_x.CFL$,$P_y.CFL$) for k-itemset algorithm.

Input: CFL-structures of $P_x.CFL$ and $P_y.CFL$.
Output: CFL-structure of $P_{xy}.CFL$.

if *x, y is generated from the same item* **then**
  return ***null***.

for *each element in $P_x.CFL$* **do**
  if $\exists E_y \in P_y.iFL$ *and* $E_x.tid == E_y.tid$ **then**
    $E_{xy} \leftarrow (E_x.tid,\ min(E_x.fv,\ E_y.fv),\ min(E_x.rmrfv,\ E_y.rmrfv))$;
    $P_{xy}.CFL \leftarrow P_{xy}.CFL + E_{xy}$.

return $P_{xy}.CFL$.

An example is given below to show the process for how to construct the CFL-structure. For example, the CFL-structure of $(b.M,\ e.H)$ is constructed having four elements $(4, 0.13, 0)$, $(5, 0.13, 0.13)$, $(6, 0.56, 0)$ and $(8, 0.56, 0)$, which is shown in Fig. 4. The element $(4, 0.13, 0)$ is constructed from elements $(4, 0.13, 0.13)$ and $(4, 0.13, 0)$ as: $(4, min(0.13, 0.13), min(0.13, 0)) = (4,\ 0.13,\ 0.13)$.

After the CFL-structure is generated, we then present another pruning strategy to reduce the size of the search space by using the Sup and $rSup$ of such a list X to decide whether to search the extension of X. The strategy is described as Lemma 2.

Lemma 2 For a termset X, if $Sup(X)$ or relative remaining support $rSup(X)$ is less than the minimum support threshold, then any supersets (extension) of X is not a MFFP and should be discarded.

b.M, e.H		
4	0.13	0
5	0.13	0.13
6	0.56	0
8	0.56	0

b.M, c.H		
5	0.13	0.13
9	0.56	0.56

b.M, e.L		
4	0.13	0
5	0.13	0
9	0.56	0

b.M, e.M		
4	0.13	0
5	0.13	0
6	0.56	0
8	0.56	0
9	0.56	0

Fig. 4: CFL-structures for the 2-itemsets.

Algorithm 3: Developed EFM algorithm.

Input: $CFLs$, the built CFL-structure.
Output: $MFFPs$, the set of multiple fuzzyfrequent patterns.

for *each list* X *in* $CFLs$ **do**
  if $Sup(X) \geq minSup$ **then**
    add items of X into $MFFPs$;
    if $rSup(X) \geq minSup$ **then**
      $exCFLs \leftarrow null$;
      for *each iFL-structure* Y *after* X *in* $CFLs$ **do**
        $exCFLs \leftarrow exCFLs + Constrcut(X, Y)$;
      EFM($exCFLs$);
return $F2FPs$.

The developed EFM algorithm is then shown in Algorithm 3. First, the algorithm begins with the initially constructed CFL-structures, and for each termset (such as X), the $Sup(X)$ is firstly compared with the $minSup$ to examine whether X is frequent. After that, the relative remaining support value of X, called $rSup(X)$, is then utilized to decide whether the extensions of X should be explored. Here, a construction function in Algorithm 2 is then performed to build the extensions of the termset X. After that, the algorithm is processed again for the next k-itemsets until all the required MFFPs are determined.

5 Experimental Evaluation

In this section, the performance of the developed EFM is then compared to the level-wise algorithm [7] and list-based approach [44] in several known datasets. The algorithms were implemented using the popular `JAVA` language, performing on a PC with Intel Core i7-3470@3.40GHz and 8GB main RAM. All of the algorithms as implemented are programmed and administered on a 64-

bit Microsoft Windows 10 OS (Operating System). We use six real-world [1] chess, retail, foodmart, mushroom, and BIBLE datasets, and one synthetic T10I4D100K dataset were conducted for all experiments. The parameters are stated as follows. #|**D**| is the size of transactions in the database; #|**I**| represents the number of items, and each item is a distinct item to others; **AvgLen** is the average value of transaction length, and **MaxLen** shows maximal length value of the transactions. Furthermore, the characteristics of the conducted datasets are shown in Table 6.

Table 6: Characteristics of used datasets.

Dataset	#|**D**|	#|**I**|	**AvgLen**	**MaxLen**	**Type**
Chess	3196	75	37	37	dense
Mushroom	8,124	119	23	23	dense
Foodmart	21,556	1559	4	11	sparse
Retail	88,162	16470	10.3	76	sparse
BIBLE	36369	13,904	17	77	sparse
T10I4D100K	100,000	942	10.1	29	sparse

The purchase amount of each item in the quantitative database is first transformed according to the defined type-2 membership functions. In the experiments, the linguistic 2-terms and 4-terms respectively shown in Fig. 5 and Fig. 6 are used to show the performance of the designed model. Linguistic terms are given a user's preference.

5.1 Execution time

The execution time of the compared algorithms for 2-terms membership functions under different minimum support thresholds is first illustrated in Fig. 7.

It can be seen from the above results that the developed EFM algorithm has better execution time than the conventional level-wise and the state-of-the-art list-based algorithm for mining MFFPs with fuzzy linguistic 2-terms in all experimental datasets. From the above observation, it can be seen that the execution time decreases along with the increase of the minimum support threshold. This is acceptable since as the increasing of minimum support threshold, the number of MFFPs decreases since fewer patterns satisfy the condition with a higher threshold. For instance in Fig. 7(e), the execution times of the level-wise, list-based, and the designed EFM are respectively 389.1, 201.68, and 103.94 seconds while the minimum support threshold is set as 0.75%. When the support threshold increases to 1.05%, the execution times of the compared algorithms are 226.71, 181.45, and 95.23 seconds.

The execution times decrease with the increase of minimum support for the chess dataset mentioned in Fig. 7(a), mushroom dataset Fig. 7(b), and foodmart Fig. 7(c). The results are increased greatly to a higher ratio when

[1] https://www.philippe-fournier-viger.com/spmf/

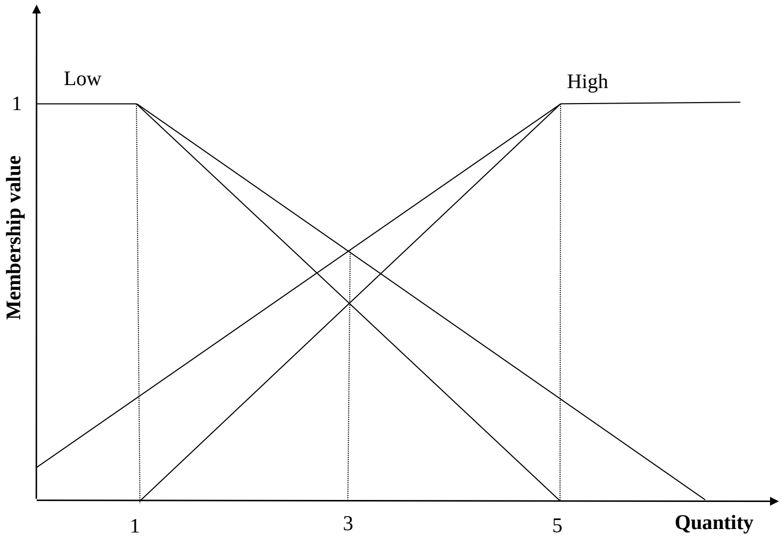

Fig. 5: The membership function of linguistic 2-terms.

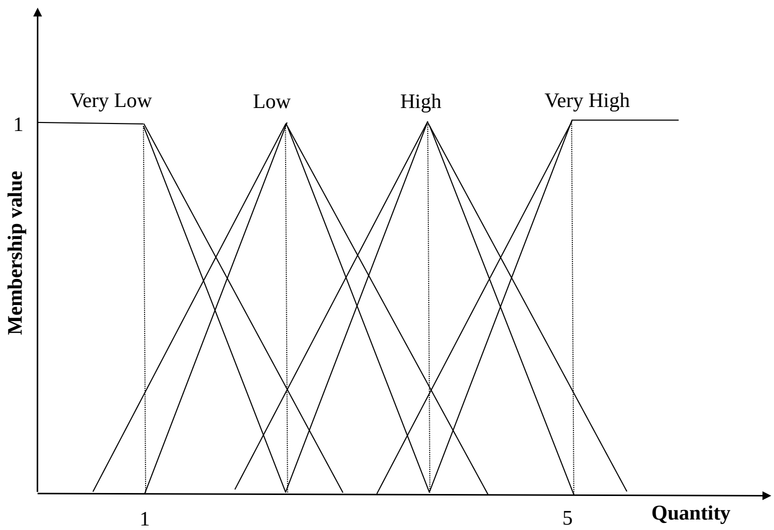

Fig. 6: The membership function of linguistic 4-terms.

compared with level-wise and list-based structure, respectively. The proposed EFM structure improvement is rational as the number of rules is decreased when the minimum threshold value is set higher. The proposed efficient compressed structure helps to reduce the runtime by ignoring certain transactions. Therefore, we can observe that the designed EFM needs fewer computations

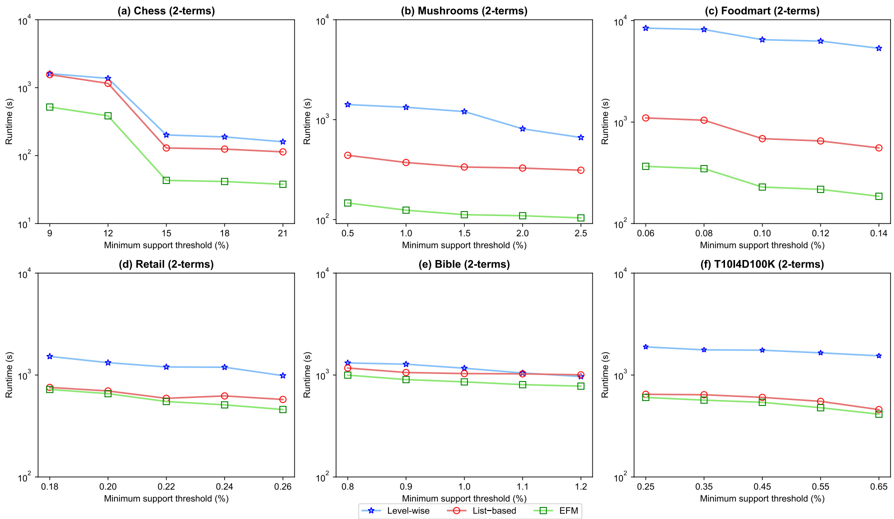

Fig. 7: Execution time comparisons with 2-terms membership functions.

than the compared approaches. Furthermore, experiments under the membership functions with linguistic 4-terms are compared and shown in Fig. 8.

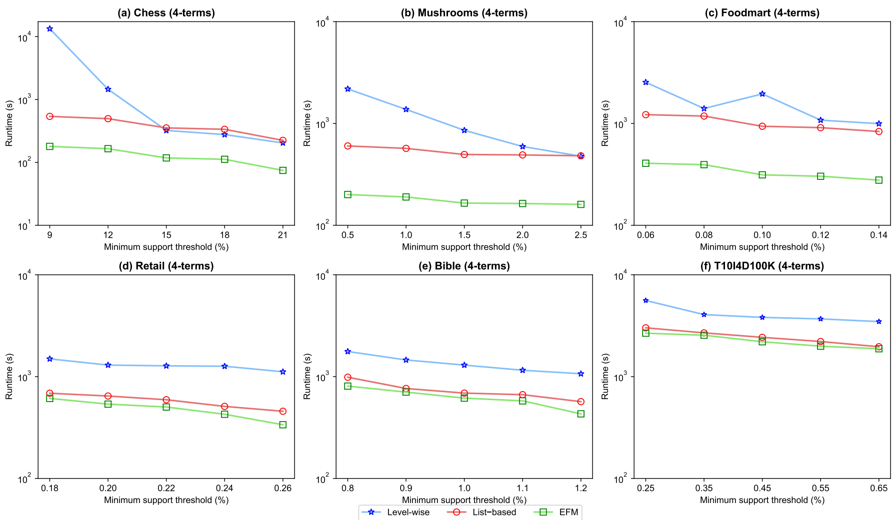

Fig. 8: Execution time comparisons with 4-terms membership functions.

In Fig. 8(a), Fig. 8(b), and Fig. 8(c) when the support values are set to low, the proposed model performed 3× better than both the list-based and level-wise algorithm. The reason is that the proposed model limits the scan for multiple transactions whereas list-based and level-wise algorithms are required to scans more transactions to extract rules. For the dense datasets, the developed EFM still performs better than compared algorithms while the threshold is set relatively low. Furthermore, the execution times of the level-wise dramatically decrease while the threshold value is set higher which can be observed in Fig. 8(a), Fig. 8(b), Fig. 8(c), Fig. 8(e), and Fig. 8(f). Thus, more execution times of the level-wise approach are required especially in the dense datasets. This is reasonable since, for every transaction in the dense datasets, it contains more items than that of the sparse ones. Thus, the developed CFL-structure can keep complete and relevant information for later progress. Furthermore, the proposed two pruning strategies are effective to reduce the size of the search space; less unpromising candidates are determined and examined compared to the level-wise and the list-based structure. Results regarding # of nodes that are examined in space (search) for the compared algorithms are then shown next.

5.2 Number of examined nodes

In this section, the number of examined nodes in the search space of the enumeration tree for the three compared algorithms are then determined. Results under linguistic 2-terms membership functions are then stated in Fig. 9.

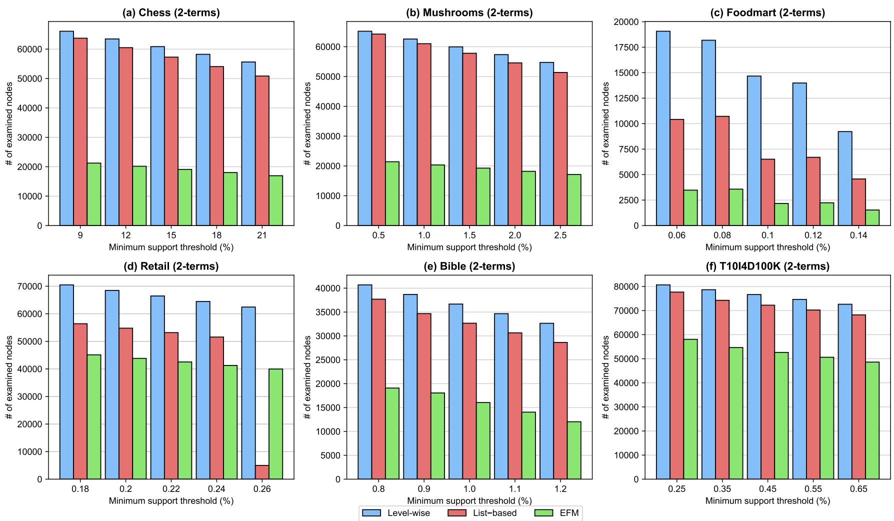

Fig. 9: Comparisons for the number of nodes under linguistic 2-terms membership functions.

In Fig. 9(a) to Fig. 9(c), it can be easily observed that the designed EFM has generated fewer nodes for examination in the search space compared to the other two approaches. The reason is that the proposed structure can efficiently reduce the number of database scans by keeping relevant information. Therefore, the proposed list-based structure reduces the search space size by limiting the extraction rules. For instance in Fig. 9(f), the level-wise and the list-based approaches respectively need to examine $872,334$ and $870,801$ candidates but the developed EFM only examines $869,203$ for the actual $2,837$ MFFPs while the minimum support threshold is set as 0.2%. When the threshold increases, for example, 0.50%, the level-wise and the list-based methods required to examine $396,025$ and $396,017$ nodes respectively but the EFM approach determines $333,853$ candidates when the threshold is set as 0.45%. We can also observe that the difference between the compared algorithms is not huge from Fig. 9(f). The reason is that this dataset belongs to the sparse dataset; the relevant relationship of the items in the database is thus low. Besides, the examined nodes in the search space are not considered as the MFFPs; many candidates are determined but fewer patterns are considered as the MFFPs. Thanks to the advantage of the designed two pruning strategies, they are effective to reduce some unpromising candidates for examination in the search space of the MFFPs. Experiments for the linguistic 4-terms membership functions are then conducted and shown in Fig. 10.

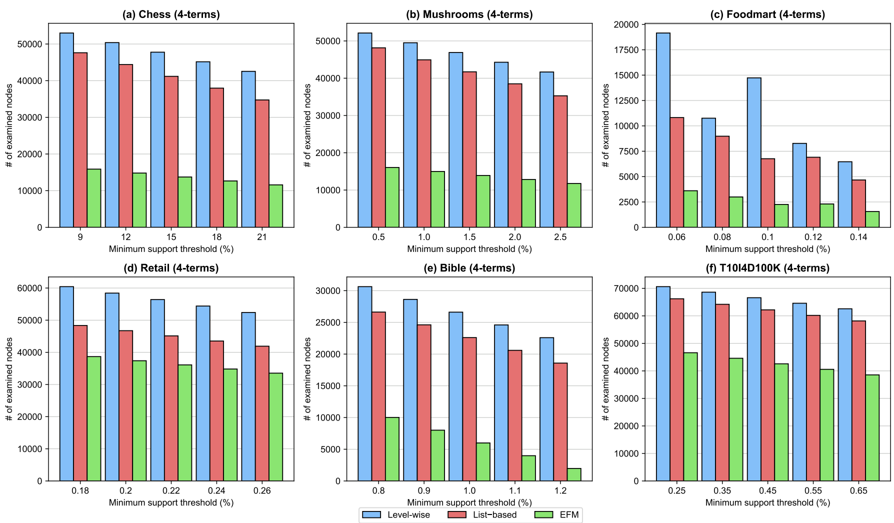

Fig. 10: Comparisons for the number of nodes under linguistic 4-terms membership functions.

Generally, the designed EFM performs better than the compared level-wise and list-based approaches, especially in Fig. 10(e), the number of examined nodes for the level-wise approach is almost more than twice of the developed EFM approach. The reason is in this dataset, the produced linguistic terms are highly relevant; the designed pruning strategies are effective to reduce the size of the unpromising patterns in the search space, and the list-based approach keeps a little bit more nodes for an examination compared to the designed EFL-structure. Also, it still can be found that the designed EFL-structure is better than the past list-based approach, which can be seen in Fig. 10(c) while the minimum support threshold is set higher (from 0.13% to 0.18%). Furthermore, in Fig. 10(f), it can be observed that the three compared algorithms showed almost the same size as the determined nodes. The reason is that for this sparse dataset, since it is hard to find the relevant information of the determined linguistic terms, thus the pruning strategies do not well perform to early reduce the number of examined candidates; the compared algorithms almost produce the same size as the determined nodes in the search space.

5.3 Memory usage

In this section, the Java API is used to measure the memory usage for the compared algorithms under six databases. Results are then shown in Fig. 11 and Fig. 12 respectively for 2-terms and 4-terms membership functions.

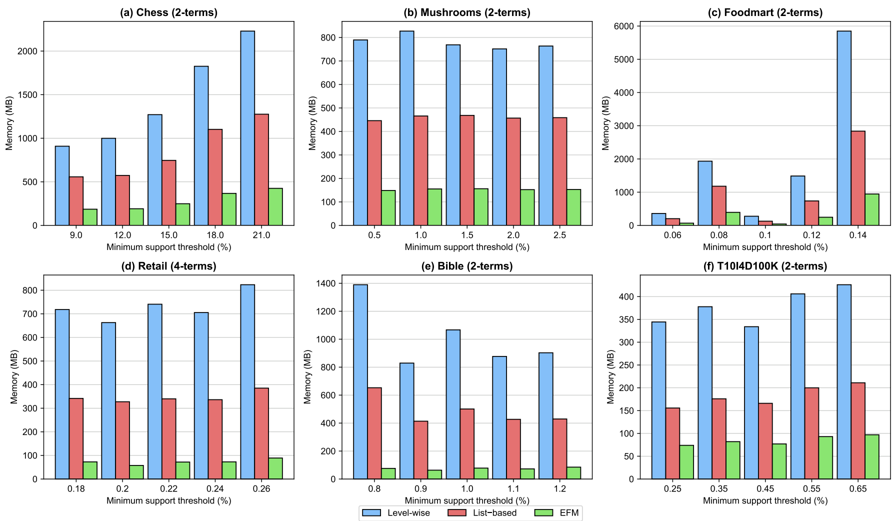

Fig. 11: Comparisons for the memory under linguistic 2-terms membership functions.

From the results, it can be seen that the designed EFM algorithm requires less memory usage compared to the level-wise and the list-based models. As the increasing of the threshold value, the designed EFM remains stable for the memory usage, as well as the list-based algorithm except in the foodmart database with 2-terms membership functions shown in Fig. 11(c). Moreover,

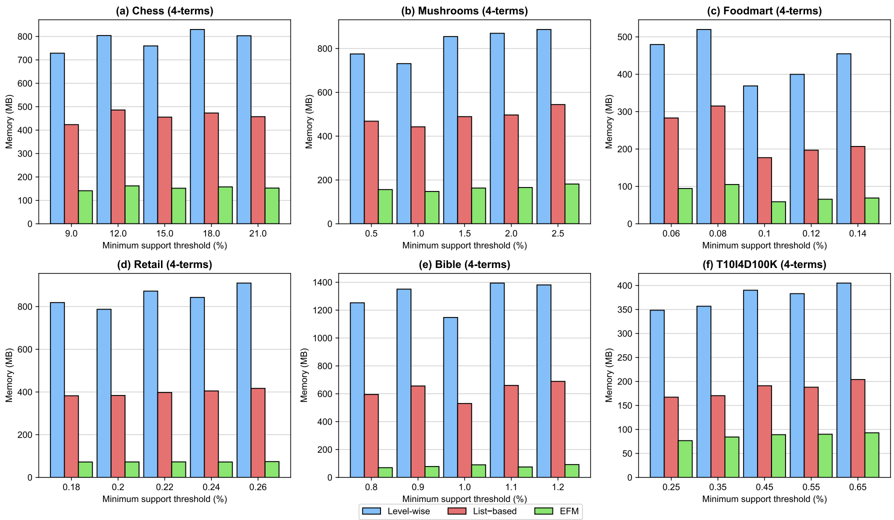

Fig. 12: Comparisons for the memory under linguistic 4-terms membership functions.

the level-wise algorithm requires the most memory usage since it needs to perform the multiple database scans for the generate-and-test mechanism. In general, the designed pruning strategies used in the EFM model can efficiently reduce memory usage than the state-of-the-art approaches. For both membership terms, the proposed algorithm performed 3× better. The memory usage is reduced due to the filtration of a large number of unpromising patterns. When the support threshold value is set to low, the level-wise and list-based algorithms required more time to search the required information, which makes it harder to return extracted rules.

5.4 Scalability

In this subsection, the proposed algorithm is compared to the state-of-the-art algorithms in terms of memory usage under 2-terms and 4-terms membership functions. Experiments are then performed under synthetic `T10I4N4KDXK` dataset. The dataset with various number of transactions X (from $100k$ to $500k$, increments $100k$ each time) was generated using the simulated IBM Quest synthetic data generator [2]. During experiments, we set the utility threshold as 20%. The results are compared in terms of runtime, memory consumption and the number of visited nodes shown in Fig. 13(a) to Fig. 13(d), respectively.

From the scalability analysis, it is observed that the proposed algorithm always performs better in terms of runtime, memory usage, and the visited

[2] http://www.Almaden.ibm.com/cs/768 quest/syndata.html

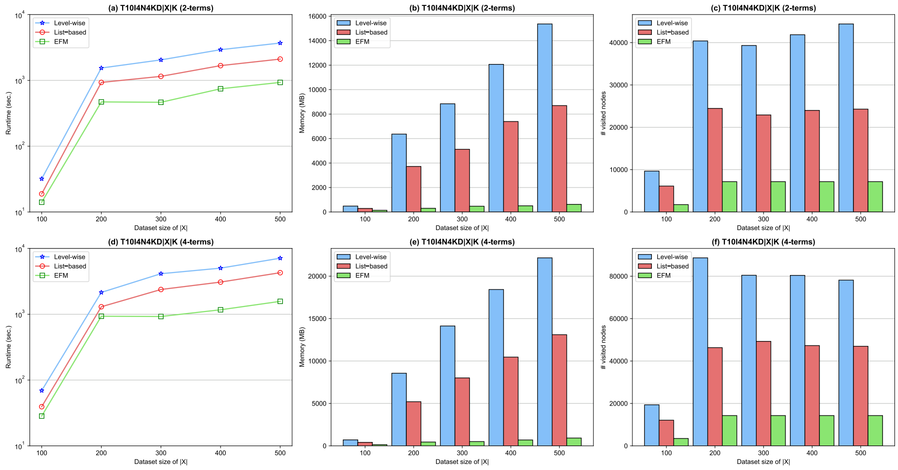

Fig. 13: Scalability results.

number of nodes compared to the other given approaches. The level-wise algorithm has the most runtime, memory consumption, and the number of visited candidates for mining the MFFIs. The list-based algorithm has better results than that of the level-wise approach. This is reasonable since the list-based algorithm can reduce the computational cost for multiple database scans. However, the designed model utilized an improved list structure and efficient pruning strategies, thus reducing the memory usage and the number of visited candidates. Moreover, the proposed algorithm follows the linear trend when the transaction size is increased from $100k$ to $500k$. The observed linear trend suggests that the proposed algorithm has excellent performance and scalable for handling large datasets. From the results, it can be concluded that the proposed algorithm has good robustness and more scalable to handle the big data issue compared to the state-of-the-art approaches.

6 Conclusions and Future Works

In this paper, an efficient fuzzy mining (EFM) algorithm is presented to discover the set of multiple fuzzy frequent patterns (MFFPs) based on the type-2 fuzzy-set theory. A compressed fuzzy-list (CFL) is also maintained for storing the satisfied fuzzy frequent itemsets that reduce the conventional limitation of multiple database scans. Two effective pruning strategies are also designed to reduce the unpromising candidates early, thus reduces the search space to find the required MFFPs. Experiments were performed on six datasets varying minimum thresholds to verify the performance of the designed EFM method compared to the previous two works in terms of execution time and the number of examined nodes in the search space. In the future, a more condensed structure and tighter upper-bound values should be explored on patterns to speed up the mining processes' efficiency. Moreover, it is also a big challenge to maintain sufficient information for incremental mining in dynamic databases or efficiently synthesizing the discovered knowledge (i.e., MFFPs) from different branches which should be considered in further studies.

References

1. R. Agrawal, T. Imieliński, and A. Swami, *Mining association rules between sets of items in large databases*, ACM SIGMOD Record, 1993, pp. 207–216.
2. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," The International Conference on Very Large Databases, pp. 487–499, 1994.
3. W. H. Au and K. C.C. Chan, "An effective algorithm for discovering fuzzy rules in relational databases," IEEE International Conference on Fuzzy Systems, pp. 1314–1319, 1998.
4. M. S. Chen, J. Han, and P. S. Yu, "Data mining: An overview from a database perspective," IEEE Transactions on Knowledge and Data Engineering, vol. 6, pp. 866–883, 1996.
5. C. Li, B. Yan, M. Tang, J. Yi, and X. Zhang, "Data driven hybrid fuzzy model for short-term traffic flow prediction," Journal of Intelligent & Fuzzy Systems, vol. 35, pp. 6525–6536, 2018.
6. J. S. Chen, F. G. Chen, and J. Y Wang, "Enhance the multi-level fuzzy association rules based on cumulative probability distribution approach," The ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 89–94, 2012.
7. C. H. Chen, T. P. Hong, and Y. Li, "Fuzzy association rule mining with type-2 membership functions," Lecture Notes in Computer Science, pp. 128–134, 2015.
8. W. Gan, J. C. W. Lin, P. Fournier-Viger, H. C. Chao, V. S. Tseng, and P. S. Yu, "FDHUP: Fast algorithm for mining discriminative high utility patterns," Knowledge and Information Systems, vol. 51(3), pp. 873–909, 2017.
9. P. K. Gupta and P. K. Muhuri, "Perceptual reasoning based solution methodology for linguistic optimization problems," `https://arxiv.org/abs/2004.14933`, 2020.
10. J. Holland, "Adaptation in natural and artificial systems," Cambridge, MA: MIT Press, 1975.
11. J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," The International Conference on Very Large Data Bases, pp. 420–431, 1995.
12. T. P. Hong, C. S. Kuo, and S. C. Chi, "Mining association rules from quantitative data," Intelligent Data Analysis, vol. 3, pp. 363–376, 1999.
13. J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent Patterns without candidate generation: a frequent-pattern tree approach,", Data Mining & Knowledge Discovery, vol. 8, 53–87, 2004.
14. H. Hagras, *Type-2 fuzzy logic controllers: A way forward for fuzzy systems in real world environments*, Lecture Notes in Computer Science, 2008, pp. 181–200.
15. T. P. Hong, G. C. Lan, Y. H. Lin, and S. T. Pan, "An effective gradual data-reduction strategy for fuzzy itemset mining," International Journal of Fuzzy Systems, vol. 15(2), pp. 170–181, 2013.
16. T. P. Hong, C. W. Lin, and T. C. Lin, "The MFFP-tree fuzzy mining algorithm to discover complete linguistci frequent itemsets," Computational Intelligence, vol. 30, pp. 145–166, 2014.
17. N. N. Karnik and J. M. Mendel, "Introduction to type-2 fuzzy logic systems," International Conference on Fuzzy Systems, pp. 915–920, 1998.
18. C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," ACM SIGMOD Record, vol. 27, pp. 41–46, 1998.
19. S. Kar and M. M. J. Kabir, "Comparative analysis of mining fuzzy association rule using genetic algorithm," The International Conference on Electrical, Computer and Communication Engineering, pp. 1–5, 2019

20. C. W. Lin, T. P. Hong, and W. H. Lu, *The Pre-FUFP algorithm for incremental mining*, Expert Systems with Applications, **36** (2009), 9498–9505.
21. C. W. Lin, T. P. Hong, and W. H. Lu, "Linguistic data mining with fuzzy FP-trees," Expert Systems with Applications, vol. 37, pp. 4560–4567, 2010.
22. C. W. Lin, T. P. Hong, and W. H. Lu, "An efficient tree-based fuzzy data mining approach," International Journal of Fuzzy Systems, vol. 12, pp. 150–157, 2010.
23. C. W. Lin and T. P Hong, *A survey of fuzzy web mining*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 3, pp. 190–199, 2013.
24. C. W. Lin and T. P. Hong, "Mining fuzzy frequent itemsets based on UBFFP trees," Journal of Intelligent and Fuzzy Systems, vol. 27, pp. 535–548, 2014.
25. J. C. W. Lin, T. P. Hong, and T. C. Lin, "A CMFFP-tree algorithm to mine complete multiple fuzzy frequent itemsets," Applied Soft Computing, vol. 28, pp. 431–439, 2015.
26. J. C. W. Lin, T. P. Hong, T. C. Lin, and S. T. Pan, "An UBMFFP tree for mining multiple fuzzy frequent itemsets," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 23, pp. 861–879, 2015.
27. J. C. W. Lin, T. Li, P. Fournier-Viger, and T. P. Hong, "A fast algorithm for mining fuzzy frequent itemsets," Journal of Intelligent & Fuzzy Systems, vol. 29, pp. 2373–2379, 2015.
28. J. C. W. Lin, X. Lv, P. Fournier-Viger, T. Y. Wu, and T. P. Hong, "Efficient mining of fuzzy frequent itemsets with type-2 membership functions," The Asian Conference on Intelligent Information and Database Systems, pp. 191–200, 2016.
29. J. C. W. Lin, L. Yang, P. Fournier-Viger, J. M. T. Wu, T. P. Hong, L. S. L. Wang, and J. Zhan, "Mining high-utility itemsets based on particle swarm optimization," Engineering Applications of Artificial Intelligence, vol. 55, pp. 320–330, 2016.
30. P. Fournier-Viger, C. W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," Data Science and Pattern Recognition, vol. 1, pp. 54–77, 2017.
31. J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and H. C. Chao, "Mining of skyline patterns by considering both frequent and utility constraints," Knowledge and Information Systems, vol. 51(3), pp. 873–909, 2017.
32. J. C. W. Lin, G. Srivastava, Y. Zhang. Y. Djenouri, and M. Aloqaily, "Privacy preserving multi-objective sanitization model in 6G IoT environments," IEEE Internet of Things Journal, 2020.
33. J. C. W. Lin, Y. Shao, Y. Djenouri, and U. Yun, "ASRNN: A recurrent neural network with an attention model for sequence labeling," Knowledge-based Systems, 2020.
34. J. M. Mendel, and R. I. B. John, "Type-2 fuzzy sets made simple," IEEE Transactions on Fuzzy Systems, vol. 10, pp. 117–127, 2002.
35. D. Mishra, S. Mishra, S. K. Satapathy, and S. Patnaik, "Genetic algorithm based fuzzy frequent pattern mining from gene expression data," Soft Computing Techniques in Vision Science, pp. 1–14, 2012.
36. R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," The SIGMOD International Conference on Management of Data, pp. 1–12, 1996.
37. A. K. Shukla and P. K. Muhuri, "Big-data clustering with interval type-2 fuzzy uncertainty modeling in gene expression datasets," Engineering Applications of Artificial Intelligence, vol. 77, pp. 268–282, 2019.
38. D. K. Srivastava, B. Roychoudhury, and H. V. Samalia, "Fuzzy association rule mining for economic development indicators," International Journal of Intelligent Enterprise, vol. 6(1), pp. 3–18, 2019.
39. G. Srivastava, J. C. W. Lin, X. Zhang, and Y. Li, "Large-scale high-utility sequential pattern analytics in Internet of things," IEEE Internet of Things Journal, 2020.
40. G. Srivastava, J. C. W. Lin, A. Jolfaei, Y. Li, and Y. Djenouri, "Uncertain-driven analytics of sequence data in IoCV environments," IEEE Transactions on Intelligent Transportation Systems, 2020.
41. T. Watanabe and R. Fujioka, "Fuzzy association rules mining algorithm based on equivalence redundancy of items," IEEE International Conference on Systems, Man, and Cybernetics, pp. 1960–1965, 2012.
42. J. M. T. Wu, J. C. W. Lin. and A. Tamrakar, "High-utility itemset mining with effective pruning strategies," ACM Transactions on Knowledge Discovery from Data, vol. 13, Article 58, 2019.

43. L. Wang, Q. Ma, and J. Meng, "Incremental fuzzy association rule mining for classification and regression," IEEE Access, vol. 7, pp. 121095–121110, 2019.
44. T. Y Wu, J. C. W. Lin, U. Yun, C. H. Chen, G. Srivastava, and X. Lv, "An efficient algorithm for fuzzy frequent itemset mining," Journal of Intelligent & Fuzzy Systems, pp. 1–11, 2020.
45. L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, pp. 338–353, 1965.

Appendix:

Lemma 1: For an termset X, if $Sup(X)$ or $rSup(X)$ is less than the minimum support threshold, then any supersets (extension) of X is not multiple fuzzy frequent pattern and should be pruned.

Proof $\forall$ transaction $T \supseteq X'$,
$\because X'$ is an extension of X, $(X' - X) = (X'/X)$, we can obtain that $X \subseteq X' \subseteq T \Rightarrow (X'/X) \subseteq (T/X)$,
$\therefore fv(X', T) = fv(X, T) \cup fv((X' - X), T) = min(fv(X, T), fv(X'/X, T)) \leq fv(X, T)$ and $min(fv(X, T), fv(X'/X, T)) \leq fv(X'/X, T) = rmrfv(X, T)$.

Suppose that $X.tids$ denotes the set of *tids* of X,
$\because X \subseteq X' \Rightarrow X'.tids \subseteq X.tids$,
$\therefore \frac{\sum_{id(T) \in X'.tids} fv(X',T)}{N} \leq \frac{\sum_{id(T) \in X.tids} fv(X,T)}{N} \Rightarrow Sup(X) < minSup$.

Furthermore, we can obtain that $\frac{\sum_{id(T) \in X'.tids} rmrfv(X',T)}{N} \leq \frac{\sum_{id(T) \in X.tids} rmrfv(X,T)}{N} \Rightarrow rSup(X) < minSup$.

Lemma 2: For a termset X, if $Sup(X)$ or relative remaining support $rSup(X)$ is less than the minimum support threshold, then any supersets (extension) of X is not a MFFP and should be discarded.

Proof $\because X \subseteq X' \Rightarrow X'.tids \subseteq X.tids$,
$\therefore Sup(X') = \frac{\sum_{id(T) \in X.tids} fv(X',T)}{N} = \frac{\sum_{id(T) \in X'.tids} min(fv(X,T), fv(X'/X,T)}{N}$
$\leq \frac{\sum_{id(T) \in X'.tids} min(fv(X,T), rmrfv(X,T)}{N} = \frac{\sum_{id(T) \in Q'} fv(X,T) + \sum_{id(T) \in Q''} rmrfv(X,T)}{N} = rSup(X) \leq minSup$.

Note that suppose $Q' \cup Q'' = X'.tids$ and $Q' \cap Q'' = $, $T \in Q'$, $fv(X, T) < rmrfv(X, T)$, and $T \in Q'$, $fv(X, T) \geq rmrfv(X, T)$.