# An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining

A Thesis submitted to Gujarat Technological University

For the Award of

**Doctor of Philosophy**

In

**Computer/IT Engineering**

By

**Patel Mahendrakumar Narottamdas**
Enrollment No. 189999913028

Under the supervision of

**Dr. Sanjay M. Shah**

**GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD**

**March - 2024**

# An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining

A Thesis submitted to Gujarat Technological University

For the Award of

**Doctor of Philosophy**

in

**Computer/IT Engineering**

By

**Patel Mahendrakumar Narottamdas**
Enrollment No. 189999913028

Under the supervision of

**Dr. Sanjay M. Shah**

**GUJARAT TECHNOLOGICAL UNIVERSITY,
AHMEDABAD**

**March - 2024**

# DECLARATION

I declare that the thesis entitled **"An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining,"** submitted by me for the degree of Doctor of Philosophy, is the record of research work carried out by me during the period from February 2019 to December 2023 under the supervision of **Dr. Sanjay M. Shah** and this has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of Research Scholar: Date: 4/3/24

Name of Research Scholar: **Patel Mahendrakumar Narottamdas**

Place: GTU, Ahmedabad

# CERTIFICATE

I certify that the work incorporated in the thesis "**An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining**" submitted by **Mr. Patel Mahendrakumar Narottamdas** was carried out by the candidate under my supervision/guidance. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship, or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of Supervisor: Date: 4|3|'24

Name of Supervisor: **Dr. Sanjay M Shah**

Place: GTU, Ahmedabad

# Course-work Completion Certificate

This is to certify that Mr. **Patel Mahendrakumar Narottamdas,** Enrollment no.**189999913028**, is a PhD scholar enrolled for the PhD program in the Computer/IT Engineering branch of Gujarat Technological University, Ahmedabad.

**(Please tick the relevant option(s))**

□ He has been exempted from the course-work (successfully completed during M.Phil. Course)

□ He has been exempted from Research Methodology Course only (successfully completed during M.Phil Course)

☑ He has successfully completed the Ph.D. course work for the partial requirement for the award of Ph.D. Degree. His performance in the course work is as follows-

| Grade Obtained in Research Methodology (PH001) | Grade Obtained in Self Study Course (Core Subject) (PH002) |
|---|---|
| BC | AB |

Supervisor's Sign

(Dr. Sanjay M. Shah)

# Originality Report Certificate

It is certified that Ph.D. Thesis titled **"An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining"** by **Patel Mahendrakumar Narottamdas** has been examined by us.
We undertake the following:

a. Thesis has significant new work/knowledge as compared to already published or is under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph, or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. There is no plagiarism). No ideas, processes, results, or words of others have been presented as the Author own work.

c. There is no fabrication of data or results which have been complied/analysed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using **"Drillbit Plagiarism Checker"** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index <=10 %).

Signature of Research Scholar: Date: 4/3/'24

Name of Research Scholar: **Patel Mahendrakumar Narottamdas**

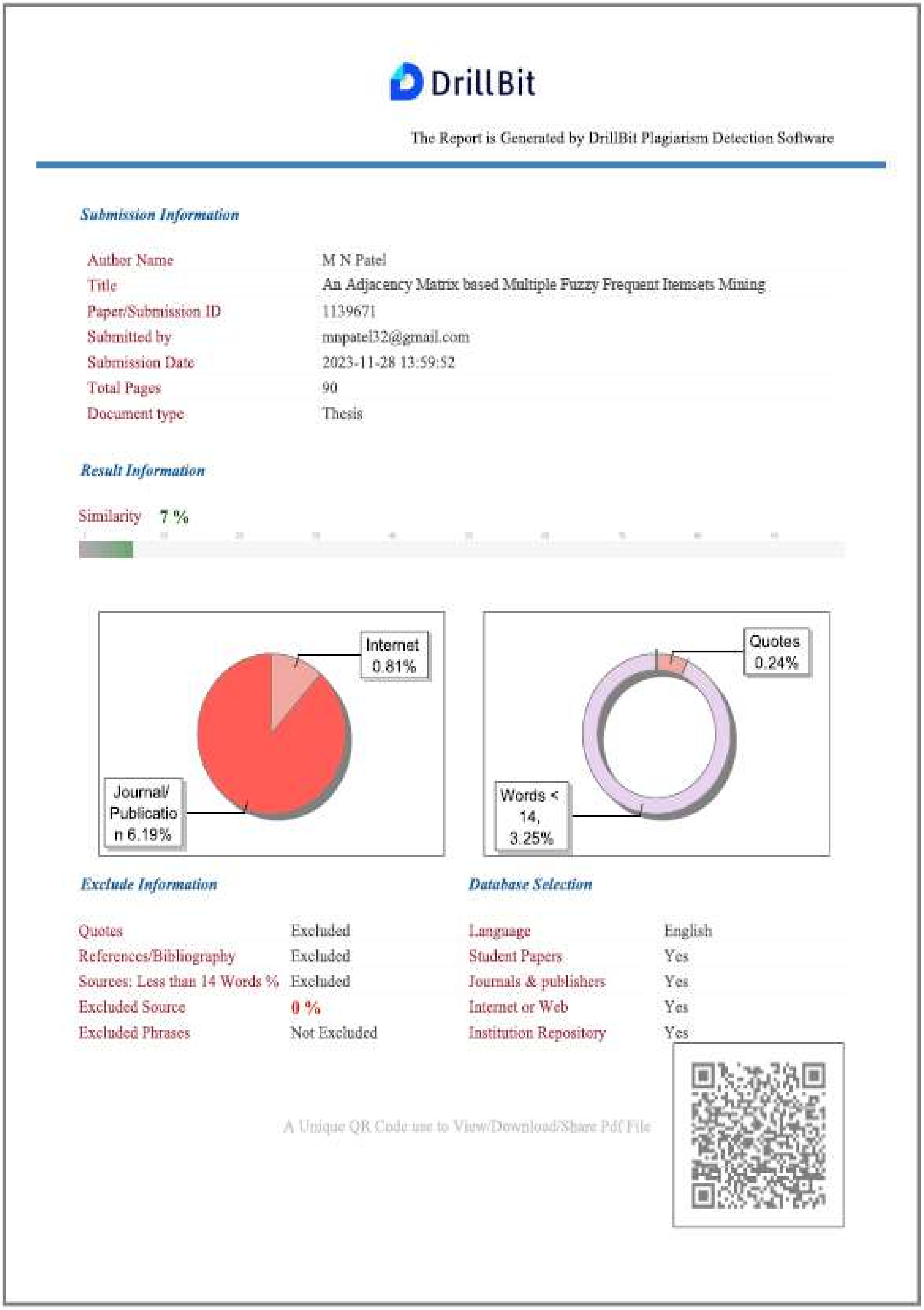
Place: GTU, Ahmedabad

Signature of Supervisor: Date: 4/3/'24

Name of Supervisor: **Dr. Sanjay M. Shah**

Place: GTU, Ahmedabad

# Copy Originality Report



DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

***Submission Information***

| | |
|---|---|
| Author Name | M N Patel |
| Title | An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining |
| Paper/Submission ID | 1139671 |
| Submitted by | mnpatel32@gmail.com |
| Submission Date | 2023-11-28 13:59:52 |
| Total Pages | 90 |
| Document type | Thesis |

***Result Information***

Similarity 7 %

Internet 0.81%

Journal/ Publication 6.19%

Quotes 0.24%

Words < 14, 3.25%

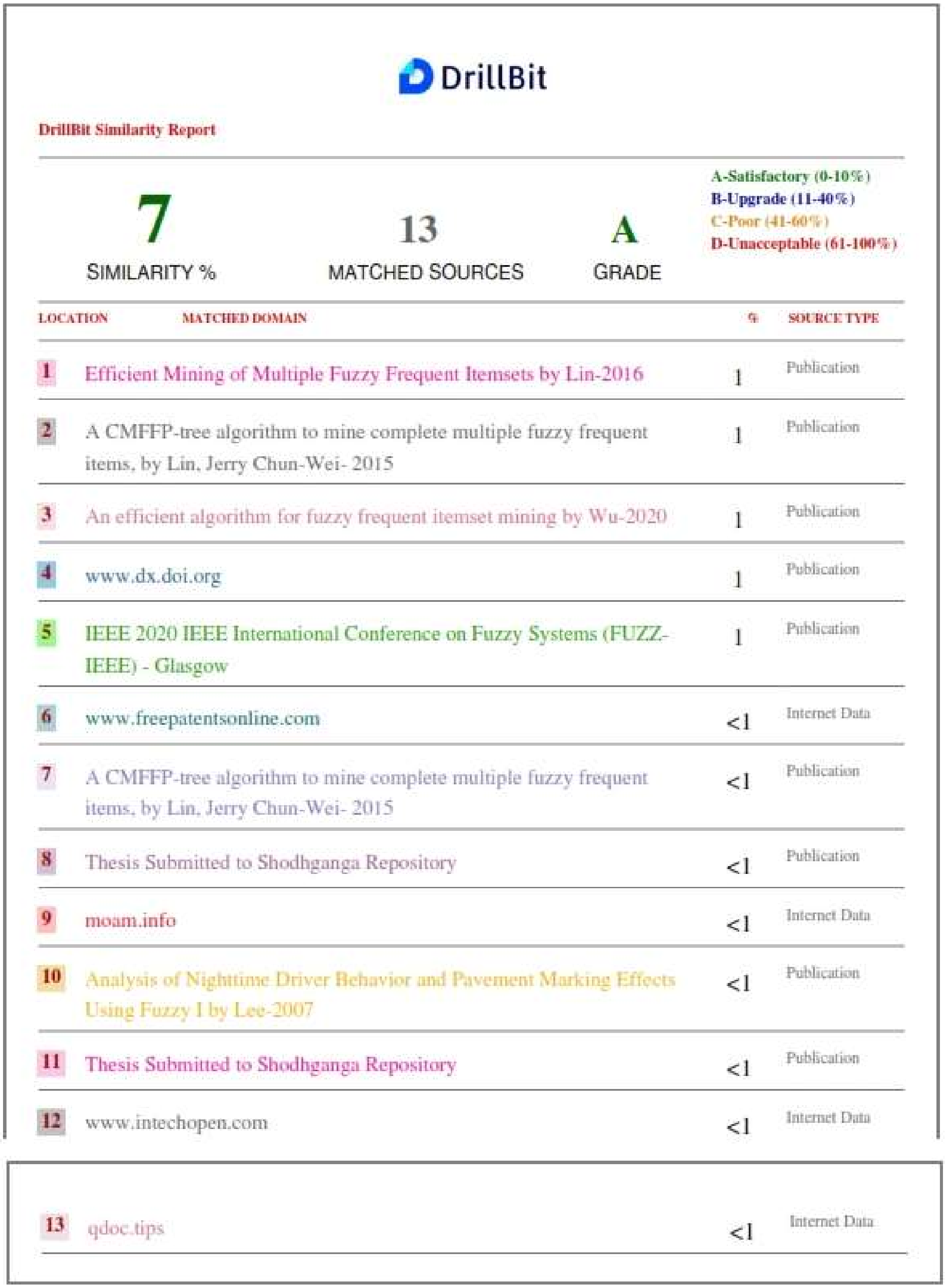
***Exclude Information***

| | |
|---|---|
| Quotes | Excluded |
| References/Bibliography | Excluded |
| Sources: Less than 14 Words % | Excluded |
| Excluded Source | 0 % |
| Excluded Phrases | Not Excluded |

***Database Selection***

| | |
|---|---|
| Language | English |
| Student Papers | Yes |
| Journals & publishers | Yes |
| Internet or Web | Yes |
| Institution Repository | Yes |

A Unique QR Code use to View/Download/Share Pdf File



DrillBit

DrillBit Similarity Report

| 7 | 13 | A | A-Satisfactory (0-10%)<br>B-Upgrade (11-40%)<br>C-Poor (41-60%)<br>D-Unacceptable (61-100%) |
|---|---|---|---|
| SIMILARITY % | MATCHED SOURCES | GRADE | |

| LOCATION | MATCHED DOMAIN | % | SOURCE TYPE |
|---|---|---|---|
| 1 | Efficient Mining of Multiple Fuzzy Frequent Itemsets by Lin-2016 | 1 | Publication |
| 2 | A CMFFP-tree algorithm to mine complete multiple fuzzy frequent items, by Lin, Jerry Chun-Wei- 2015 | 1 | Publication |
| 3 | An efficient algorithm for fuzzy frequent itemset mining by Wu-2020 | 1 | Publication |
| 4 | www.dx.doi.org | 1 | Publication |
| 5 | IEEE 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) - Glasgow | 1 | Publication |
| 6 | www.freepatentsonline.com | <1 | Internet Data |
| 7 | A CMFFP-tree algorithm to mine complete multiple fuzzy frequent items, by Lin, Jerry Chun-Wei- 2015 | <1 | Publication |
| 8 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 9 | moam.info | <1 | Internet Data |
| 10 | Analysis of Nighttime Driver Behavior and Pavement Marking Effects Using Fuzzy I by Lee-2007 | <1 | Publication |
| 11 | Thesis Submitted to Shodhganga Repository | <1 | Publication |
| 12 | www.intechopen.com | <1 | Internet Data |
| 13 | qdoc.tips | <1 | Internet Data |

# PhD Thesis Non-Exclusive License to
# GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being PhD Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere I, "**Patel Mahendrakumar Narottamdas**" having Enrollment No. **189999913028** hereby grant a non-exclusive, royalty-free and perpetual license to GTU on the following terms:

a. The University is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part ( referred to collectively as the "Work") anywhere in the world, for non-commercial purposes, in all forms of media;

b. The University is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);

c. The University is authorized to submit the Work at any National / International Library, under the authority of their "Thesis Non-Exclusive License";

d. The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;

e. I undertake to submit my thesis, through my University, to any Library and Archives. Any abstract submitted with the thesis will be considered to form part of the thesis.

f. I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.

g. If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

h. I understand that the responsibility for the matter as mentioned in the paragraph (g) rests with the authors / me solely. In no case shall GTU have any liability for any acts / omissions / errors / copyright infringement from the publication of the said thesis or otherwise.

i. I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.

j. GTU logo shall not be used /printed in the book (in any manner whatsoever) being published or any promotional or marketing materials or any such similar documents.

k. The following statement shall be included appropriately and displayed prominently in the book or any material being published anywhere: "The content of the published work is part of the thesis submitted in partial fulfilment for the award of the degree of Ph.D. in Computer/IT Engineering of the Gujarat Technological University".

l. I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this nonexclusive license. I shall keep GTU indemnified from any and all claims from the Publisher(s) or any third parties at all times resulting or arising from the publishing or use or intended use of the book / such similar document or its contents.

m. I am aware of and agree to accept the conditions and regulations of Ph.D. including all policy matters related to authorship and plagiarism.

Date: 4/3/'24

Place: GTU, Ahmedabad

Signature of the Research Scholar

Recommendation of the Research Supervisor: RECOMMENDED

Signature of the Research Supervisor

# Thesis Approval Form

The viva-voce of the PhD Thesis submitted by **Mr. Patel Mahendrakumar Narottamdas** (Enrollment No. **189999913028**) entitled " **An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining"** was conducted on 04th March-2024 at Gujarat Technological University.

**(Please tick any one of the following options)**

☑ The performance of the candidate was satisfactory. We recommend that he be awarded the PhD degree.

□ Any further modifications in research work recommended by the panel after 3 months from the date of first viva- voce upon request of the Supervisor or request of Independent Research Scholar after which viva – voce can be re-conducted by the same panel again.

**(Briefly specify the modifications suggested by the panel)**

□ The performance of the candidate was unsatisfactory. We recommend that he should not be awarded the PhD degree.

**(The panel must give justifications for rejecting the research work)**

| | |
|---|---|
| (Dr. Sanjay M. Shah) | (Dr. Nilesh R. Patel) |
| Name and signature of Supervisor with Seal | External Examiner-1 (Name and Signature) |

(Dr. Dharm Singh Jat)
External Examiner-2 (Name and Signature)

# Abstract

Discovering helpful information from transactions is becoming an important research issue. Several frequent itemsets mining algorithms are proposed for association rule mining, which handle only binary datasets. These methods concentrate on an item's presence or absence in a dataset. However, in some situations in real life, it is crucial to consider the quantity of items. A fuzzy technique is used to handle quantitative datasets and to generate meaningful representations of the dataset. Thus several algorithms were developed to discover fuzzy frequent itemsets from quantitative transactions. Most of them merely take the linguistics term with the highest cardinality into account. As a result, the number of original elements and fuzzy regions processed is equal. On the other hand, decision-making can be made more successful when an item has several fuzzy zones.

Existing approaches scan the database more than once, and the high number of join counts (candidate itemsets) required thus degrade the algorithm's performance by increasing execution time.

In this research, we proposed AMFFI (Adjacency matrix based multiple fuzzy frequent itemsets mining) and MFFPA-2 (multiple fuzzy frequent itemsets mining using adjacency matrix with type-2 membership function) using an Adjacency matrix and Fuzzy-Tid-list structures to discover multiple fuzzy frequent itemsets (MFFI) that scan the database only once.

AMFFI is proposed for mining MFFI from quantitative transactions. AMFFI technique uses a type-1 membership function to transform quantitative datasets into fuzzy linguistics terms. An efficient search space exploration strategy is proposed to find the occurrence of two fuzzy linguistic terms together immediately from the adjacency matrix to minimize the join counts and speed up the discovery of MFFI.

The proposed MFFPA-2 uses a type-2 membership function to transform quantitative databases into fuzzy linguistics terms. The type 2 Fuzzy Set could be useful for providing more reliable and agile decision-making by considering many uncertainty possibilities and considering more complex relationships between variables.

Extensive experiments have been conducted to verify efficiency regarding runtime, memory usage, and join counts with different min support thresholds. Experimental results demonstrate that the designed approaches AMFFI and MFFPA-2 achieved superior performance compared to cutting-edge techniques. The AMFFI improves execution time by 8% to 81% and node join count by 93% to 99%. The MFFPA-2 improves execution time by 38% to 75% and node join count by 93% to 99%.

# Acknowledgment

I sincerely thank everyone who has supported and encouraged me during my doctoral research journey.

I want to begin my heartfelt thanks to Dr. Sanjay M Shah, my Ph.D. supervisor. He is a Professor and Head of the Computer Engineering Department at Government Engineering College, Rajkot. Sir has consistently provided me with invaluable support and thoughtful guidance throughout the entire duration of my research. I am profoundly grateful for his unique insights, continuous motivation, and the significant time he dedicated to shaping this research and illuminating a previously undisclosed aspect of the study.

I want to extend my sincere gratitude to my Doctorate Progress Committee (DPC) members, namely Dr. Chirag S. Thaker, who serves as a Head and Professor in the Computer Engineering Department at L.D. College of Engineering, Ahmedabad, and Dr. Sanjay P. Patel, an Assistant Professor in the Computer Engineering Department at Government Engineering College, Gandhinagar. Their insightful comments, valuable suggestions, and encouragement to view the problem from various angles have been immensely helpful. Their approachable demeanour and appreciation for my work have fostered a supportive atmosphere, boosting my confidence to overcome challenges and push my boundaries.

I am also thankful to the Honourable Vice-Chancellor, Registrar, Controller of Examination, Dean of the PhD section, and the entire team at the PhD Section of Gujarat Technological University (GTU) for their invaluable assistance and unwavering support.

I want to express my deep appreciation to Dr. S. P. Dave, the Principal of GEC Gandhinagar, Dr. D. A. Parikh, the Head of the CE Department, and Professor J. S. Dhobi, the Head of the IT department, for granting me the necessary resources and facilities to reach my desired goals.

I am grateful to my parent institution, Government Engineering College, Gandhinagar, and the Department of Technical Education, Gujarat, for their comprehensive support throughout my research journey. I am fortunate to have received blessings and guidance from Dr. K. K. Jani and Dr. Pratik Barot. Additionally, I want to express my thanks to my colleagues and well-wishers in the CE/IT department for their continuous encouragement and support during the entire duration of my research work.

I hold the utmost respect and affection for my parents, wife, brothers, son Rudra, and niece and nephews, Meshwi and Ved. I want to convey my profound gratitude for their unwavering support and collaboration. I am particularly grateful to my father, Shri Narottambhai K Patel, my mother, Puriben Patel, and my wife, Veenaben Patel, for their consistent encouragement, tireless support during my research, and cooperation.

Finally, I want to convey my profound gratitude to my dear friend Suresh B Patel from the depths of my heart. Reaching this stage would not have been possible without his continuous support, motivation, and encouragement.

Since this has been a lengthy journey, I might have unintentionally forgotten a few names, yet they have played a crucial role in this significant undertaking. I humbly ask for forgiveness for any oversight and extend my sincere respect to every one of them.

I am incredibly thankful to Almighty God for giving me the patience and strength to complete this research.

**Patel Mahendra N.**

# Table of Contents

# List of Abbreviations

| | | |
|---|---|---|
| AdjMat | : | Adjacency Matrix |
| AMFFI | : | Adjacency matrix Based Multiple Fuzzy Frequent Itemsets Mining |
| ARM | : | Association Rule Mining |
| ARs | : | Association Rules |
| CFFP | : | Compressed Fuzzy Frequent Pattern |
| CFL | : | Complex Fuzzy List |
| $C_k$ | : | k-Candidate Itemsets |
| CMFFP-tree | : | Compressed Multiple Fuzzy-Term FP-tree |
| CRM | : | Customer Relationship Management |
| DB | : | Database |
| DM | : | Data Mining |
| DSS | : | Decision Support System |
| EFM | : | Efficient Fuzzy frequent Mining |
| FCFI-miner | : | Fuzzy Closed Frequent Itemsets Miner |
| FCFI-miner | : | Fuzzy Closed Frequent Itemsets Miner |
| FFIM | : | Fuzzy Frequent Itemsets Mining |
| FFPT | : | Fuzzy Frequent-Pattern Tree |
| FI | : | Frequent Itemsets |
| FIM | : | Frequent Itemsets Mining |
| $FL_k$ | : | Fuzzy k-Frequent Itemsets |
| GDF | : | Gradual Data-Reduction Strategy For Fuzzy Itemsets Mining |
| if | : | Internal Fuzzy Value |
| IFFP | : | Improved Fuzzy Frequent Pattern Mining |
| ISPFTI | : | Integrated Sequential Pattern Mining With Fuzzy Time Intervals |
| KDD | : | Knowledge Discovery In Database |
| $L_k$ | : | k-Frequent Itemsets |
| MFFI | : | Multiple Fuzzy Frequent Itemsets |

| | | |
|---|---|---|
| MFFPA-2 | : | Multiple Fuzzy Frequent Itemsets Mining Using Adjacency matrix With Type-2 Membership Function |
| MFFP-tree | : | Multiple Fuzzy Frequent Pattern-Tree |
| min_conf | : | Minimum Confidence |
| *min_supp* | : | Minimum Support |
| NLP | : | Natural Language Processing |
| *rf* | : | Resting Fuzzy Value |
| Sup(X) | : | Support of Itemsets X |
| Tid | : | Transaction ID |
| UBFFP | : | Upper Bound Fuzzy Frequent Pattern |
| UBMFFP | : | Upper-Bound Multiple Fuzzy FP-Tree |

# List of Figures

# List of Tables

# CHAPTER 1
# Introduction

## 1.1 Introduction

In recent decades, online and mall shopping have been drastically increasing. For increasing the business, discovering valuable information from datasets is very important. Manually discovering the knowledge from the raw data is practically impossible[1][2]. Data mining is a set of techniques to discover knowledge from the raw data [3] automatically. Data mining techniques require finding knowledge from a large volume of the dataset. Association Rule mining [4][5], clustering [6], and classification [7][8] are the three primary categories of Knowledge Discovery from Dataset (KDD) methods [4]. Frequent itemsets mining (FIM) is a fundamental technique of data mining and association rule discovery, which aims to identify patterns or associations in large datasets. These patterns, referred to as "frequent itemsets," represent sets of items (e.g., products in a store, words in a document, or genes in a biological dataset) that frequently occur together in the data. FIM is also used to study customer buying patterns from historical data in market basket analysis [9]. FIM identifies a grouping of items commonly bought together by analyzing transaction datasets. A common occurrence in transaction databases involves the identification of frequent itemsets, which comprise items regularly purchased by customers in numerous transactions[10]. The exploration of frequent itemsets constitutes an extensively researched task in data mining, with its applications spanning various domains. Essentially, it entails the analysis of a database to unveil instances where specific values (items) co-occur within a collection of database entries (transactions) [11]. An item's frequency may be less exciting for users as it shows only the number of times the item appears in the dataset [12].

In FIM, consider only item presence, not the quantity of items. Quantitative databases are used in frequent itemsets mining for decision-making in real-world scenarios. It is challenging to manage the quantitative database. Mostly, all authors used fuzzy set theory to manage quantitative databases.

### 1.1.1 Need for Fuzzy Techniques in Frequent Itemset Mining:

While exploring frequent itemsets holds significance across various domains, conventional methods face constraints when handling real-world data characterized by uncertainty, vagueness, and imprecision.

**Uncertain Data:** In various applications, data might need to neatly align with binary categories. For example, in healthcare, the presence of a symptom may be uncertain. The fuzzy set theory allows us to model and analyse such uncertainty.

**Partial Membership:** Fuzzy sets permit elements to have degrees of membership, which better represent the real world. Some items or symptoms may partially belong to an itemset, reflecting their gradual relevance.

**The granularity of Data:** Fuzziness allows for a more granular representation of data, capturing nuances that binary approaches cannot. This is crucial in applications where fine distinctions matter.

**Robustness:** Fuzzy frequent itemset mining can be more robust to noise and outliers in data. It can find meaningful patterns even when there is a moderate level of uncertainty in the data.

**Complex Relationships:** Real-world associations can be complicated and subtle. Fuzzy techniques can reveal intricate relationships among items, helping us understand data more deeply.

**Improved Decision-Making:** Fuzzy frequent itemset mining provides richer insights into data, which can lead to better decision-making in various fields, including healthcare, marketing, and finance.

In summary, fuzzy techniques are essential in frequent itemsets mining to handle the inherent uncertainty and vagueness of real-world data. They empower precise and nuanced identification of patterns, allowing the extraction of valuable insights from data that conventional binary methods may fail to recognize.

## 1.2 Problem Statement

As per studies concluded by Jerry Lin et al. (2017) [13] fuzzy-list based, and in 2020 [14], compressed fuzzy list-based level-wise MFFI approaches are recent and better performed than other approaches. However, the performance of the fuzzy-list-based approaches is limited due to multiple-time database scanning and performing several costly fuzzy-list node join operations.

The focused problem statement of this research is:
*"Design an efficient and accurate method to generate multiple fuzzy frequent itemsets using Adjacency matrix."*

To achieve this we need to design a method which scans database only once and minimizes the node join counts.

## 1.3 Research Aim and Objectives

This research aims to optimize running time, memory usage, and the number of node joins required in fuzzy frequent itemsets mining. This research work proposes to achieve the following objectives:

- To study and investigate existing methods for fuzzy frequent itemsets mining.
- To recognize or determine the challenges for the fuzzy frequent itemsets mining.
- Identifying opportunities for enhancing the performance of the fuzzy frequent itemsets mining methods.
- To create and explore an effective method for navigating search spaces, aiming to decrease the expenses associated with fuzzy list join operations by minimizing the number of comparisons needed to join fuzzy lists.
- To design a novel structure that can store the fuzzy value of the itemsets to develop an efficient pruning mechanism.
- To develop and investigate an efficient pruning mechanism to diminish the number of join operations by eliminating unnecessary join operations of fuzzy lists.
- To evaluate performance and compare the results with existing state-of-the-art methods.

**Research Hypothesis**: An optimization of running time, memory usage, and number of node join counts.

## 1.4 Scope

The scope of research in fuzzy frequent itemsets mining is broad and encompasses various aspects of data mining, fuzzy logic, and practical applications. Here, we present a summary of the research scope within this particular field:

**Algorithm Development:** Creating effective algorithms for extracting fuzzy frequent itemsets from extensive datasets and exploring optimization methods to enhance the speed and scalability of mining algorithms.

**Fuzzy Membership Functions:** Investigating various forms of fuzzy membership functions for representing the extent to which items belong to itemsets. Analyze the impact of different membership functions on mining results and performance.

The research focused on transactional static datasets.

## 1.5 Significance of the Fuzzy Frequent Itemsets Mining in various Applications

### 1. Enhanced Pattern Discovery:

**Relevance:** Fuzzy frequent itemsets mining extends the capabilities of traditional frequent itemsets mining by accommodating uncertain, imprecise, and vague data. This holds significant importance in fields characterized by inherently ambiguous data, such as healthcare, natural language processing, and image analysis.

**Impact:** Researchers and practitioners can discover more meaningful patterns in data, leading to improved decision-making, enhanced predictions, and deeper insights. For example, in medical diagnosis, fuzzy itemsets mining can help identify subtle symptom-disease associations that are not seen in binary data.

**2. Improved Recommendation Systems:**

**Relevance:** Recommender systems rely on understanding user preferences, which are often uncertain and fuzzy. Fuzzy frequent itemsets mining can better capture user behaviour and preferences.

**Impact:** Enhanced recommendation accuracy can increase user satisfaction and engagement, potentially boosting sales and user retention for businesses like e-commerce platforms and streaming services.

**3. Healthcare and Biomedical Research:**

**Relevance:** Healthcare and genomics data often contain uncertainties, variations, and imprecise measurements. Fuzzy frequent itemsets mining is relevant for discovering subtle relationships between symptoms, diseases, genetic markers, and treatment responses.

**Impact:** Potential outcomes include improved disease diagnosis, personalized treatment plans, and drug discovery. Fuzzy techniques can contribute to advancements in precision medicine.

**4. Natural Language Processing (NLP):**

**Relevance:** Text data in NLP applications can be inherently fuzzy, with word senses, sentiment, and meaning varying degrees. Fuzzy frequent itemsets mining can identify nuanced patterns in textual data.

**Impact:** More accurate sentiment analysis, topic modelling, and text summarization can be achieved, improving information retrieval and content recommendation.

**5. Marketing and Customer Insights:**

**Relevance:** In market basket analysis and customer behaviour studies, fuzzy frequent itemsets mining is crucial when dealing with uncertain purchasing behaviors and product preferences.

**Impact:** Companies can enhance their inventory management processes, cross-selling strategies, and targeted marketing campaigns, ultimately increasing revenue and customer satisfaction.

**6. Environmental Monitoring:**

**Relevance:** Environmental data often contains uncertainty, especially in sensor readings and climate data. Fuzzy frequent itemsets mining can uncover complex relationships in these datasets.

**Impact:** A better understanding of environmental factors can increase effectiveness in climate change modelling, pollution control, and disaster prediction.

**7. Fraud Detection:**

**Relevance:** Deceptive actions might display patterns that are challenging to identify in clear-cut data. Fuzzy techniques can help identify irregular and ambiguous patterns in financial transactions.

**Impact:** Enhanced fraud detection and reduced false positives can save financial institutions significant resources and protect customers from fraudulent activities.

**8. Social Network Analysis:**

**Relevance:** In social network analysis, relationships among individuals can be nuanced and uncertain. Fuzzy frequent itemsets mining can uncover more realistic social structures.

**Impact:** Improved community detection, influence analysis, and recommendation systems can enhance understanding of social networks and user behaviour.

In conclusion, fuzzy frequent itemsets mining research has the potential to significantly impact data mining and related fields by enabling the extraction of valuable knowledge from fuzzy and uncertain data. Its relevance extends to a wide range of applications, from healthcare to marketing to environmental monitoring, where the richness and nuances of fuzzy patterns can lead to more informed decisions and better outcomes.

## 1.6 Contribution

The main contribution of this research is to design an efficient multiple fuzzy frequent itemsets mining method using an adjacency matrix. Proposed methods an Adjacency matrix based Multiple Fuzzy Frequent Itemsets mining (AMFFI) and Multiple Fuzzy Frequent Patterns Mining with Adjacency matrix and Type-2 member function (MFFPA-2) scan the database only once and reduce the number of node join counts (candidate itemsets) by pruning un-frequent itemsets extracted from the adjacency matrix.

The research methodology comprises developing a novel approach to mining MFFIs using an adjacency matrix and fuzzy-tid-list structures—performance of the proposed procedures AMFFI and MFFPA-2 evaluated with state-of-the-art methods on standard real datasets.

## 1.7 Organization of the Thesis

The succeeding Chapter 2 describes preliminaries, which include the concept of data mining and fuzzy frequent itemsets mining, Chapter 3 consists of a literature review, Chapter 4 shows a proposed method (AMFFI), Chapter 5 shows performance evaluation of AMFFI, Chapter 6 shows another proposed method MFFPA-2, Chapter 7 shows performance evaluation of MFFPA-2, and Chapter 8 discusses the conclusions and future enhancement at the end there are references.

# CHAPTER 2
# Background Study of Data Mining

## 2.1 Data Mining

Due to globalization and the open market period, today's businesses struggle to survive under rigorous competition [15]. In today's business landscape, companies produce vast amount of data on their products, clientele, sales, manufacturing, consumption, expenditures, and more [16][17]. The corporate entity should leverage the examined data to create diverse decision support systems for manufacturing, sales, customer relationships, and others, aiming to establish a dominant presence in the market [18]. Examining these extensive volumes of data manually is practically unfeasible. Data analysis requires the use of automatic techniques. When analyzing data to find relevant knowledge, data mining is essential. Data mining is the tools and techniques to discover knowledge from massive raw data automatically. These methods strive to find patterns not previously identified [19]. Data mining is alternatively recognized as Knowledge Discovery from Data (KDD) [19]. Data mining has three techniques: Clustering, Classification, and Association Rules mining [19][20].

- **Classification: -** Assigns items in a collection to specific categories or classes to precisely predict the target class for each instance using the provided data [21].

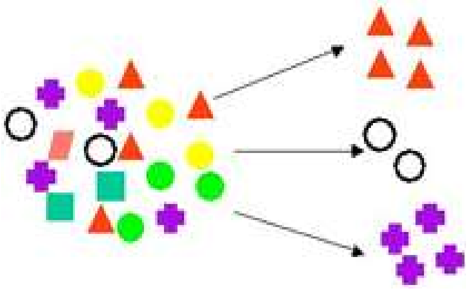


Figure 2.1: Classifications

- **Clustering: -** Organizing items into categories based on the information provided in the data that delineates the characteristics of the items or their relationships [22].

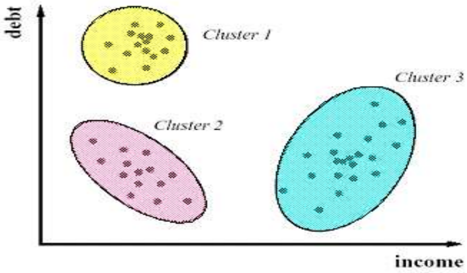


Figure 2.2: Clustering

- **Association Rules: -** One way to determine the relationship between the itemsets is through association rule mining. The frequency with which the itemset is connected to others [23].

### 2.1.1 Knowledge Discovery From Database

The increased focus on information mining in the data industry can be attributed to the widespread availability of large volumes of data and the essential need to convert such data into valuable information and knowledge. The concept of information mining is a natural progression within the field of information technology. Information mining plays a crucial role as a fundamental stage in exploring knowledge within databases. The process of knowledge discovery involves an iterative series of subsequent steps [5]:

1. Data cleansing involves the elimination of noise and inconsistent data.
2. Integration of data involves the combination of multiple sources of data.
3. Data selection involves retrieving pertinent information from the database for analysis.
4. Data transformation occurs through summarization or aggregation to prepare for mining.
5. Data mining is a crucial step that employs intelligent methods to extract patterns from the data.
6. Evaluation of patterns aims to identify noteworthy patterns that signify knowledge based on specific measures.
7. Presentation of knowledge involves utilizing visualization and representation techniques to convey mined knowledge to the user.

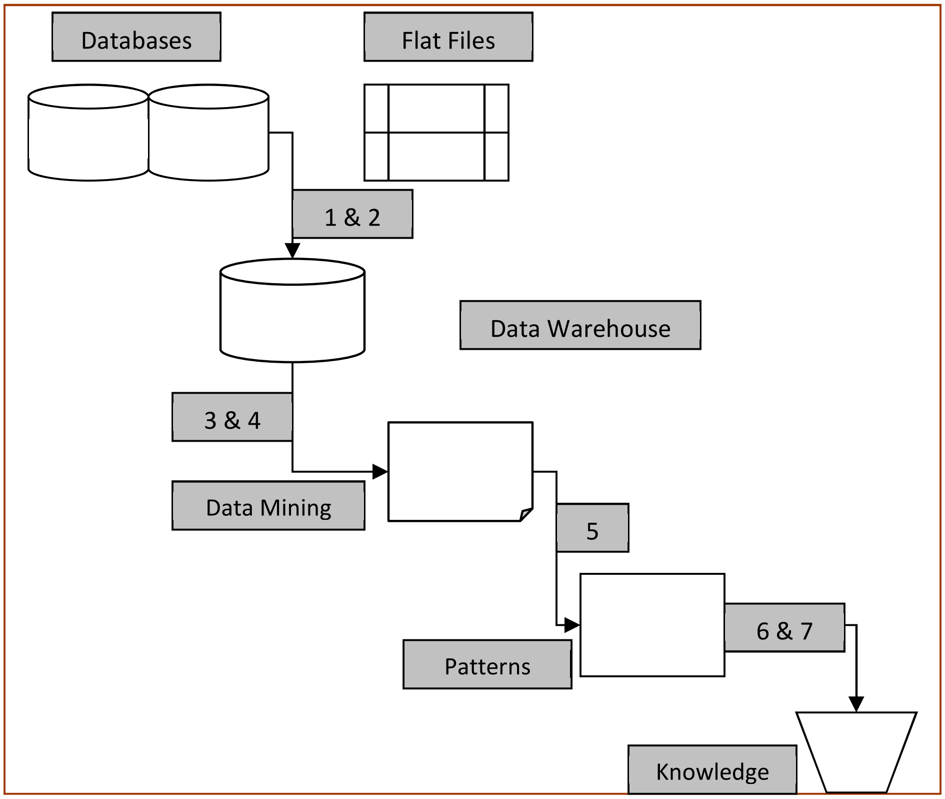


Figure 2.3: Data mining as a step in the process of knowledge discovery [5]

The inception of data mining traces back to when corporate data was initially stored in computers, and subsequent technologies were developed to enable users to interact with the data in real-time.

Data mining is becoming increasingly prevalent in both private and public sectors. Various industries, such as retail, insurance, healthcare, and banking, commonly employ data mining techniques to minimize expenses, enhance research efforts, and boost sales. In the public sector, the initial applications of data mining were primarily focused on detecting fraud and waste; however, they have also expanded to include purposes such as measuring and enhancing program performance.

Data mining can be applied to subjective, printed, or multimedia information. Utilizing data mining applications facilitates the analysis of this information. These applications encompass association (identifying patterns where one event is linked to another, such as purchasing milk and buying butter), sequence or path analysis (detecting patterns where one event leads to another, like the birth of a child and a subsequent purchase), classification (recognizing new ways, like correlations between duct tape purchases and plastic sheeting acquisitions), clustering (identifying and grouping sets of unidentified facts, such as geographical location and brand preferences), and forecasting (discerning patterns to make reasonable predictions about future activities, for instance, predicting that individuals who join an athletic club may enroll in exercise classes).

### 2.1.2 Association Rule Mining

Extensive amount of data undergo analysis through association rule mining to reveal captivating connections and associations. This regulation indicates the frequency of occurrences of an itemset within a given dataset [24]. Association rule mining can identify the correlation among the items [5]. It is a two-step procedure; in the initial phase, it identifies the frequent pattern from the extensive data and generates interesting association rules in the later step [25] for the given set of items and transactions in the retail dataset. The transaction contains a collection of items. The expression of the association rules A → B where A and B are the set of items. The rule indicates that the transactions that include itemset A tend to have itemset B [26][27].

#### 2.1.2.1 Frequent Itemsets Mining

The initial stage in association rule mining involves frequent itemset mining, which identifies a collection of items that commonly coexist in a transaction database. The frequency is determined by the number of transactions that include the itemset. The occurrence rate of the itemset is more than the user-defined *min_supp* threshold; the itemsets are frequent itemsets [28].

**Definition 1: support count** (σ)

The support count of itemset is defined as the number of transactions containing itemset.

$$\sigma\,(X) = |\ \{\, T_i \mid X \subseteq T_i,\ T_i \ \in T \,\}\ | \qquad (2.1)$$

Where the σ defines the number of transactions $T_i$ contain itemset X. Consider the sample database as in Table 2.1. For the itemset X = {Pencil, Eraser}, the support count of X is four as pencil and eraser appear in four transactions in the dataset.

Table 2.1: Sample Database for FIM

| Transaction-ID | Items |
|---|---|
| 1 | Pen, Pencil, Eraser, |
| 2 | Pencil, Eraser, Sharpener |
| 3 | Pen, CD, DVD |
| 4 | Eraser, CD |
| 5 | Pen, CD, DVD |
| 6 | Pen, Pencil, Eraser, Sharpener |
| 7 | Pen, Eraser, Sharpener, DVD |
| 8 | Eraser, CD, DVD |
| 9 | Sharpener, CD, DVD |
| 10 | Pen, Pencil, Eraser, Sharpener, CD, DVD |

**Definition 2: Frequent Itemset (FI)**

An itemset X is said to be a frequent itemset if its support count is no less than the user-defined *min_supp* threshold.

$$\sigma\,(\mathrm{X}) \;\geq Min\_Supp \qquad (2.2)$$

For the database in Table 2.1, consider the itemset X = {pencil, eraser} and the user-defined *min_sup* is 35%. Then, the itemset X is a frequent itemset as its support count $\sigma\,(\mathrm{X})$ is 4, i.e., 40%, which is greater than *min_supp*.

### 2.1.2.2 Generation of Association Rule

Association rules describe the co-existence of the itemsets. For frequent itemset X, represented as {I1, I2, I3, … In}, A and B denote two subsets of X, A∩B = **Ø,** A ≠ **Ø,** B ≠ **Ø,** and A U B = X; then the association rule A→B holds the correlation between itemset A and B. The assessment of rule strength is determined by measures such as support and confidence [29].

**Definition 3: Support (s)**

Support of the rules specifies the count of transactions that include the set of itemsets. It is decisive how frequently the rule is relevant to a given dataset. Consider the support (s) of rule A→B, which indicates the percentage of the transactions containing A U B in the given dataset of N transactions [25].

$$s(A \rightarrow B) = \frac{\sigma\,(\mathrm{A\ U\ B})}{\mathrm{N}} \tag{2.3}$$

Consider the frequent itemset X = {pencil, eraser} discovered from the dataset in Table 1. Generate the association rule with a ***pencil → eraser***. The support (s) of the said rules is 4/10, i.e., 40% or 0.4.

**Definition 4: confidence (Conf)**

Confidence shows the reliability of the rule. Consider the confidence of rule A→B is Conf. It indicates the percentage of times A appears that also appears B. The conditional probability is P (B|A) [25].

$$Conf(A \rightarrow B) = P(B\ |A) = \frac{\sigma\,(\mathrm{A\ U\ B})}{\sigma\,(\mathrm{A})} \tag{2.4}$$

The strong association rule satisfies both the user-defined (*min_supp*) minimum support and (*min_conf*) minimum confidence. Examine the correlation between the association rule " ***pencil → eraser*** " as identified in the database in Table 2.1. The confidence Conf (***pencil → eraser)*** in every instance, if a transaction includes a pencil, it will invariably have an eraser.

Association rule mining is a well-known method in data mining to discover valuable patterns from transactional data from domains like retail stores, medical, etc. Frequent pattern mining is a subfield and the cornerstone of association rule mining. One notable constraint of frequent itemset mining (FIM) is its reliance on the binary representation of item values within transactions. It only considers the item's presence or absence. The rules derived from the FIM can guide only the co-existence of the items. FIM identified the set of items that appear frequently together in the transaction database. It does not consider the item's quantities. In real-world applications, to design an efficient decision support system, it is necessary to consider item quantities. The Fuzzy Frequent Itemsets Mining (FFIM) problem has been defined to address this FIM issue. Unlike FIM, item quantities are considered in the FFIM problem. The association rules derived from the FFIM are more significant than the FIM in designing a decision support system.

## 2.2 FIM Example

To understand the basic concepts of data mining, take the example of supermarket data. Consider a small store that sells the following items: A, B, C, D, E, F, and G of things purchased by six speculative clients, as shown in Table 2.2. Here, the row shows a transaction.

Table 2.2: Supermarket Database

| No. | Item Purchased |
|---|---|
| 1 | B, C, D, F, G |
| 2 | D, F, G |
| 3 | A, C, D, F, G |
| 4 | B, D, E, F, G |
| 5 | A, E, F, G |
| 6 | E, F, G |

1. Given a set of items, any nonempty subset is called an itemsets.
2. Given an itemsets I and a set of transactions T, the support of I concerning T, denoted by support T (I), is the number of transactions in T that contain all the items in I.
3. Given an itemsets I, a set T of transactions, and δ a min_supp threshold, I is a frequent itemsets if support T(I) ≥ δ.

The accompanying example delineates the ideas exhibited in the above definition.

Given S indicates the set {A, B, C, D, E, F, G} and let T mean the set of transactions that appeared in Table 2.2. Cases of itemsets are $I_1$ = {D, F, G} and $I_2$ = {A, B}. The support of itemsets $I_1$ regarding T is four since $I_1$ shows up in precisely four of the transactions in Table 2.2. The support of itemsets $I_2$ regarding T is zero because no transaction in T contains both A and B. If we set the support threshold at 3, at that point, I1 is a substantial itemset because the support of I1 is 4. Another extensive itemsets for this support threshold is $I_3$= {E, F, G}, which has a support of 3.

In example 2-FIs is {{D,F}, {D,G}, {E,F}, {E,G}, {F,G}} and 3-FIs is {{D,F,G}, {E,F,G}}. Here, frequent itemsets are finding manually based on the min_supp threshold. Manually generating FIs from large datasets is tedious, so we must use data mining techniques. The following section shows the standard classical method of frequent itemsets mining, the Apriori method.

### 2.2.1 Apriori Method

The Apriori algorithm is a significant method for extracting frequent itemsets to establish Boolean association rules. Its nomenclature is derived from its utilization of pre-existing knowledge regarding the properties of frequent itemsets. The algorithm adopts an iterative strategy called a level-wise search, in which k-itemsets are employed to investigate (k+1) itemsets. Initially, identifying the set of frequent 1-itemsets, denoted as L1, is undertaken. L1 then serves as the basis for determining L2, the collection of frequent 2-itemsets, which subsequently facilitates the identification of L3, and so forth [30][31].

Apriori method property belongs to a particular category of properties called anti-monotone in that if any itemsets are not frequent, their supersets never become frequent. The Apriori method has two main steps: join and prune.

### 2.2.2 Apriori method steps

Detailed steps of the Apriori algorithm using a sample database transaction illustrated in Table 2.3 with the algorithm steps are as follows.

Presume a minimum support threshold of 2. In the initial stage of the algorithm, every element is considered part of the collection of candidate 1-itemsets, denoted as C1. The procedure involves systematically examining all transactions to tally the frequency of each item.

Table 2.3: Sample database

| TID | List of item_IDs |
|---|---|
| T1 | 1,2,5 |
| T2 | 2,4 |
| T3 | 2,3 |
| T4 | 1,2,4 |
| T5 | 1,3 |
| T6 | 2,3 |
| T7 | 1,3 |
| T8 | 1,2,3,5 |
| T9 | 1,2,3 |

1. A minimum support threshold of 2 (equivalent to 22%, i.e., min_supp = 2/9) is provided. Identifying the set of frequent 1-itemsets, denoted as L1, follows, comprising candidate 1-itemsets that meet the minimum support criteria. The specific L1 itemsets can be found in Table 2.4.
2. To identify the collection of frequent 2-itemsets, denoted as L2, the algorithm employs the join of L1 with itself, denoted as L1*L1, to create a candidate set of 2-itemsets, referred to as C2. The operation $L_1 * L_1 = \{ 1, 2, 3, 4, 5 \} * \{ 1, 2, 3, 4, 5 \} = \{ \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3,5\}, \{4, 5\} \}$ generates total ten itemsets. Nevertheless, merely five will qualify for L2, as the residual backing falls below the stipulated minimum support threshold, shown in Table 2.5.
3. The production of the frequent 3-itemsets set, denoted as L3, is illustrated in Table 2.6. Let C3 be defined as $L_2 * L_2 = \{\{1, 2, 3\}, \{1, 2, 5\}, \{1, 3, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}\}$. Utilizing the Apriori principle, which asserts that every subset of frequent itemsets must be frequent, we can ascertain that the four subsequent candidates are excluded due to the absence of their respective subsets in L2.

4. Finally, to generate $L_4$, use $L_3$ * $L_3$ which generates $C_4$ =Φ. One 4-itemset generated is {1, 2, 3, 5}. It undergoes pruning as its subsets are only partially found in $L_3$.

Table 2.4: Frequent 1-itemset

| Itemset | Support count |
|---|---|
| {1} | 6 |
| {2} | 7 |
| {3} | 6 |
| {4} | 2 |
| {5} | 2 |

Table 2.5: Frequent 2-itemsets

| Itemset | Support Count |
|---|---|
| {1, 2} | 4 |
| {1, 3} | 4 |
| {1, 5} | 2 |
| {2, 3} | 4 |
| {2, 4} | 2 |
| {2, 5} | 2 |

Table 2.6: Frequent 3-itemsets

| Itemset | Support Count |
|---|---|
| {1, 2, 3} | 2 |
| {1, 2, 5} | 2 |

Join and Prune operations are discussed using the following example:

- Join Operation: $C_3 = L_2 * L_2$ = {{1, 2}, {1, 3}, {1, 5}, {2, 3}, {2, 4}, {2, 5}} * {{1, 2}, {1, 3} , {1, 5}, { 2, 3}, {2, 4}, {2, 5} } ={{1, 2, 3}, {1, 2, 5}, {1, 3, 5} , {2, 3, 4}, {2, 3, 5}, {2, 4, 5} } .

- Prune operation: Every frequent itemsets nonempty subsets must also be considered frequent.

  - The subsets of {1, 2, 3} are {1, 2}, {1, 3}, {2, 3}. All 2-itemsets subsets of {1, 2, 3} belong to the collection of L2. Consequently, it retains the set {1, 2, 3} in C3.

  - The subsets of {1, 3, 5} are {1, 3}, {1, 5}, {3, 5}. {3, 5} is not belongs to the collection of L2, so it never become frequent. Therefore, prune {1, 3, 5} is not included in C3.

  - After pruning, $C_3$ = {{1, 2, 3}, {1, 2, 5}}.

## 2.3 Background Study of FFIM

Quantitative databases are used in frequent itemsets mining for decision-making in real-world scenarios. It is challenging to manage the quantitative database. Mostly, all authors used fuzzy set theory to manage quantitative databases. In fuzzy set theory, quantitative values associated with an item in the transaction are transformed into linguistic terms using a pre-defined membership function [32].

Itemset is a set of items. Several ‘m’ unique items are called the itemset I ($i_1$, $i_2$,..., $i_m$). The quantitative dataset D has ‘n’ transactions made of items from itemset I, where D=$T_1$, $T_2$, $T_3$, ..., $T_n$. Every transaction contains the notation $T_q \in D$. Every transaction also includes a TID, which stands for a unique identifier. Each transaction $T_q$ consists of an item and the value of the buy quantity; let us call it $w_{iq}$. "k-itemset" refers to an itemset of length K=$i_1$, $i_2$... $i_k$.

Table 2.7 displays a sample quantitative database D of seven transactions in the example below—the minimum support δ=1. The Type-1 membership function £1 and Type-2 membership function £2 are demonstrated in Figure 2.4 and Figure 2.5, respectively.

Table 2.7: Quantitative database

| TID | Item with Quantity |
|---|---|
| 1 | A-4, B-3, C-2, D-2 |
| 2 | B-3, C-2, E-3 |
| 3 | A-5, B-3, C-4, E-4 |
| 4 | A-2, C-1, D-3 |
| 5 | A-4, B-2, C-5 |
| 6 | B-3, C-3, D-2, E-2 |
| 7 | C-3, E-2 |

Mining fuzzy-frequent itemsets typically involves the following three steps**.**

**Step 1: Determine the item's fuzzy terms (Linguistic variable).**

Consider the dataset D and item i (i⊆I), and the value of i is the collection of fuzzy terms. The built-in type-1 and type-2 membership functions produce £1 and £2, as seen in Figure 2.4 and Figure 2.5, respectively.

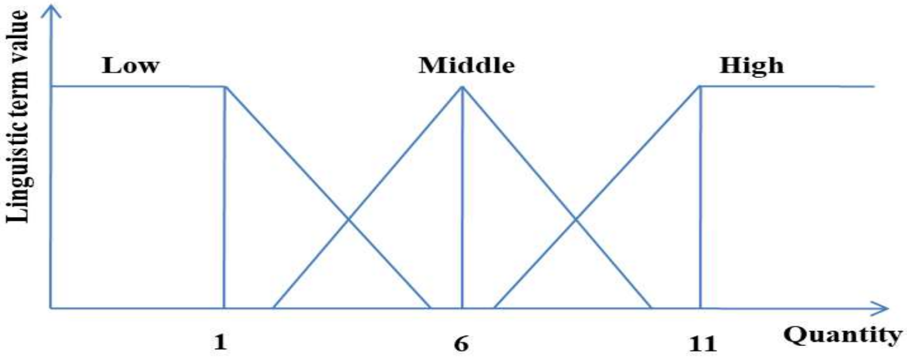


Fig. 2.4 Type-1 Membership Function

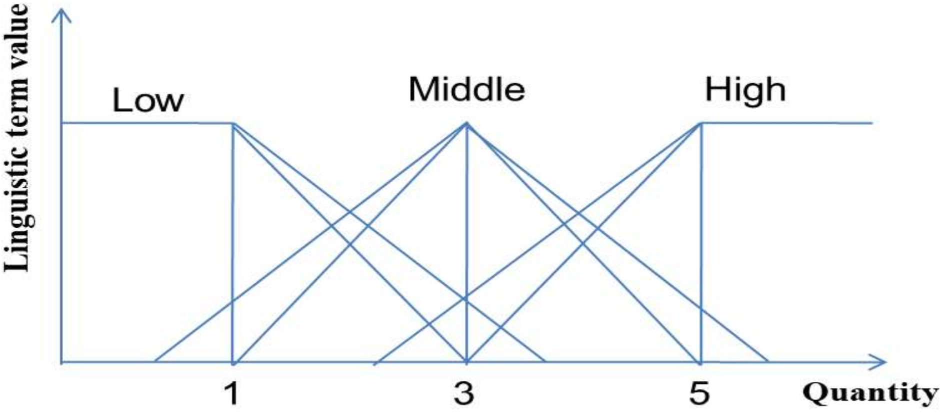


Fig. 2.5 Type-2 Membership Function

Fuzzy terms are represented as $l_{i1}$, $l_{i2}$,…, $l_{ih}$, where h is the membership degree. Figure 2.4 and Figure 2.5 show the 3-term membership function, which means here h=3. It may differ as 4-term or 5-term as per requirements. Three linguistic concepts are employed in this example: High-H, Middle-M, and Low-L. The $V_{iq}$ term represents the quantitative value of item i for transaction $T_q$, while the $F_{iq}$ term denotes the linguistic description of item i. $F_{iq}$ was generated from item i quantity value Viq using the membership function £1 or £2. Fiq for membership function £1 of $V_{iq}$ for item i is shown below.

$$f_{iq}(v_{iq}) = \frac{f_{iq1}}{l_{i1}} + \frac{f_{iq2}}{l_{i2}} \dots + \frac{f_{iqh}}{l_{ih}} \tag{2.5}$$

$F_{iqk}$ represents the fuzzy value associated with the k-th linguistic terms in $l_{ik}$, $1 \leq k \leq h$, and $f_{iqk} \subseteq [0, 1]$ [14]. For example, the membership function £1 employed in the example above represents item A with the quantity five in linguistic terms (0.2/AL, 0.8/AM, 0.0/AH). The initial step involves converting the quantitative dataset into a fuzzy set, denoted as D'. This transformation includes assigning multiple linguistic terms to each item in every transaction, as demonstrated in Table 2.8, utilizing the membership function £1.

Table 2.8: Fuzzy dataset generated by Type-1 Member function

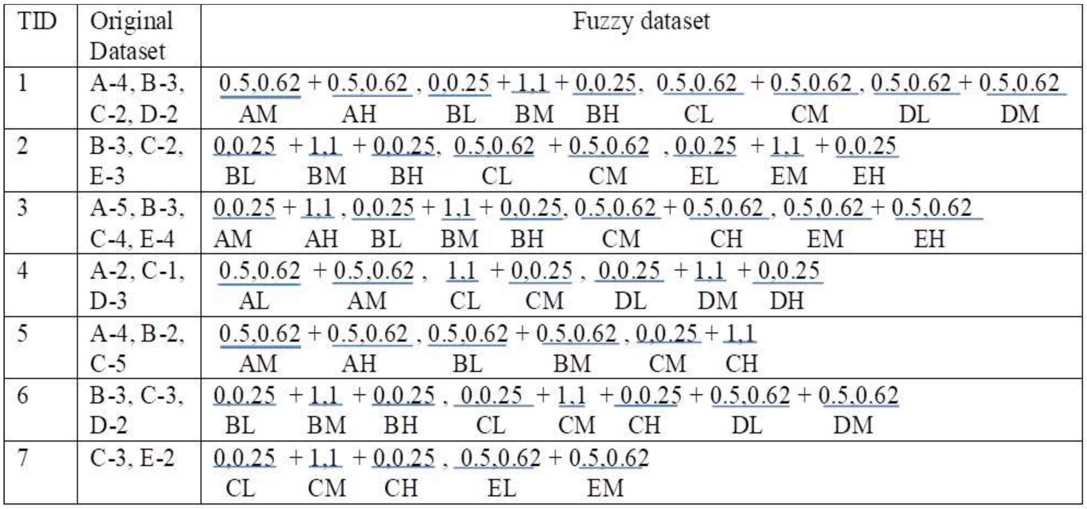
| Tid | Items | Fuzzy Linguistic terms |
|---|---|---|
| 1 | A B C D:4 3 2 2 | 0.5/AM + 0.5/AH, 1/BM, 0.5/CL + 0.5/CM, 0.5/DL + 0.5/DM |
| 2 | B C E:3 2 3 | 1/BM, 0.5/CL + 0.5/CM, 1/EM |
| 3 | A B C E:5 3 4 4 | 1/AH, 1/BM, 0.5/CM + 0.5/CH, 0.5/EM + 0.5/EH |
| 4 | A C D:2 1 3 | 0.5/AL + 0.5/AM, 1/CL, 1/DM |
| 5 | A B C:4 2 5 | 0.5/AM + 0.5/AH, 0.5/BL + 0.5/BM, 1/CH |
| 6 | B C D E:3 3 2 2 | 1/BM, 1/CM, 0.5/DL + 0.5/DM, 0.5/EL + 0.5/EM |
| 7 | C E:3 2 | 1/CM, 0.5/EL + 0.5/EM |

$F_{iq}$ for £2 is a set of three linguistic terms: $f_{iq1}^{lower}$, $f_{iq1}^{upper}/l_{i1}$ for membership value low, $f_{iq2}^{lower}$, $f_{iq2}^{upper}$ $/l_{i2}$ for membership value middle, and $f_{iq3}^{lower}$, $f_{iq3}^{upper}$ $/l_{i3}$ for membership value high as shown in following equation 2.6 [14]. Where $l_{il}$ shows l-th linguistic (fuzzy) terms, $f_{iql}^{lower}$ and $f_{iql}^{upper}$ show lower and upper membership values of $V_{iq}$ for item i.

$$f_{iq}(v_{iq}) = \frac{f_{iq1}^{lower}, f_{iq1}^{upper}}{l_{i1}} + \frac{f_{iq2}^{lower}, f_{iq2}^{upper}}{l_{i2}} \ldots + \frac{f_{iqh}^{lower}, f_{iqh}^{upper}}{l_{ih}} \quad (2.6)$$

Table 2.9 shows the resultant fuzzy dataset, say D', after applying the membership function £2 on given an example.

Table 2.9: Fuzzy dataset generated by Type-2 member function



| TID | Original Dataset | Fuzzy dataset |
|---|---|---|
| 1 | A-4, B-3, C-2, D-2 | $\frac{0.5,0.62}{AM}+\frac{0.5,0.62}{AH}, \frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CL}+\frac{0.5,0.62}{CM}, \frac{0.5,0.62}{DL}+\frac{0.5,0.62}{DM}$ |
| 2 | B-3, C-2, E-3 | $\frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CL}+\frac{0.5,0.62}{CM}, \frac{0,0.25}{EL}+\frac{1,1}{EM}+\frac{0,0.25}{EH}$ |
| 3 | A-5, B-3, C-4, E-4 | $\frac{0,0.25}{AM}+\frac{1,1}{AH}, \frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CM}+\frac{0.5,0.62}{CH}, \frac{0.5,0.62}{EM}+\frac{0.5,0.62}{EH}$ |
| 4 | A-2, C-1, D-3 | $\frac{0.5,0.62}{AL}+\frac{0.5,0.62}{AM}, \frac{1,1}{CL}+\frac{0,0.25}{CM}, \frac{0,0.25}{DL}+\frac{1,1}{DM}+\frac{0,0.25}{DH}$ |
| 5 | A-4, B-2, C-5 | $\frac{0.5,0.62}{AM}+\frac{0.5,0.62}{AH}, \frac{0.5,0.62}{BL}+\frac{0.5,0.62}{BM}, \frac{0,0.25}{CM}+\frac{1,1}{CH}$ |
| 6 | B-3, C-3, D-2 | $\frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0,0.25}{CL}+\frac{1,1}{CM}+\frac{0,0.25}{CH}+\frac{0.5,0.62}{DL}+\frac{0.5,0.62}{DM}$ |
| 7 | C-3, E-2 | $\frac{0,0.25}{CL}+\frac{1,1}{CM}+\frac{0,0.25}{CH}, \frac{0.5,0.62}{EL}+\frac{0.5,0.62}{EM}$ |

**Step 2: Find the support count of each fuzzy item.**

Fuzzy itemsets $L_{ik}$'s support count (scalar cardinality) is shown by the symbol sup ($L_{ik}$). Find each fuzzy itemset's support in this stage. According to this definition,

$$\text{Sup }(L_{ik}) = \sum_{q=0,\ L_{ik} \subseteq T_q\ ^\wedge\ T_q \in D'}^{n} f_{iqk} \quad (2.7)$$

In fuzzy dataset D', fuzzy item $L_{ik}$'s fuzzy value is $f_{iqk}$. Check each fuzzy item's Sup ($L_{ik}$); if the minimum support requirement is satisfied, place the item in $FL_1$.

$FL_1 = FL_1 \cup (\text{sup }(L_{ik}) >= \delta)$. Where $FL_1$ is fuzzy 1-frequent itemsets, and δ is the min_supp threshold.

**Step 3: Finding the Sup of each frequent fuzzy itemsets:**

The following-level frequent itemsets are fuzzy k-itemsets with k=2, produced by fuzzy 1-frequent itemsets ($FL_1$). Fuzzy items from $FL_1$ are combined using the join procedure to create a candidate set, such as $FC_2$ (fuzzy 2-candidate itemsets). Consider the itemset X that was produced by merging the $FL_1$ itemsets A and B. Sup(X) = support of itemset: X, where $X \subseteq T_q$ and $T_q \in D'$, is considered the lowest fuzzy values of fuzzy itemsets A and B from truncation $T_q$. According to this definition,

$$\text{Sup (X)} = X \in L_i \ / \sum_{q=0,\ X \subseteq T_q \ ^\wedge\ T_q \in D'}^{n} \min(\text{faq}, \text{fbq}) \qquad (2.8)$$

Store them in fuzzy 2-frequent itemsets (FL2) from FC2 itemsets if they satisfy the minimal support. Similarly, the identification of fuzzy k-frequent itemsets occurs subsequently.

- The challenge in Fuzzy Frequent Itemsets Mining (FFIM) involves identifying the entire collection of fuzzy frequent itemsets within a quantitative database:

$$\mathbf{FFIS \leftarrow \{SUP(X) \geq \delta \times |D|\},} \qquad (2.9)$$

Considering X as a fuzzy linguistic term (which can also be viewed as itemsets), sup(X) represents the support count of X, where δ denotes the minimum support threshold, and |D| indicates the size of the quantitative database.

# Chapter 3

# Literature Review

Detecting fuzzy frequent itemsets in transaction datasets continues to pose a significant challenge, even with continuous advancements over the past two decades. Figure 3.1 depicts the publication trends of different approaches to FFIMS from 2001 to 2022. The publication data was sourced from Google Scholar, using the search query "fuzzy frequent itemsets mining". From the statistical fact, it has been observed that FFIM is the demanded research topic in the research community.

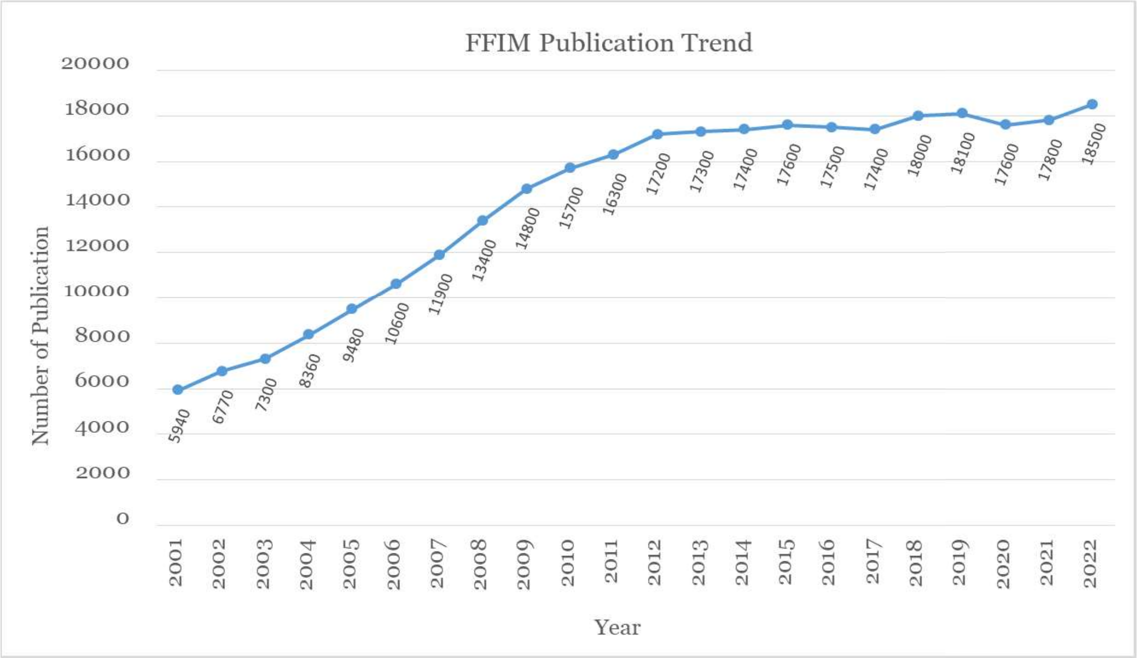


Fig. 3.1: FFIM publication trends

## 3.1 Introduction

It is essential to extract and present information from real-world data in an understandable format. Linguistic representation is popular because it makes information more accessible for people to understand. The representation can be effortlessly accomplished using fuzzy set theory because fuzzy set theory involves natural language for quantification and reasoning. Methods for mining fuzzy frequent itemsets can be categorized into horizontal and pattern growth algorithms. Horizontal (Level-wise) approach generates patterns containing 1 item, then 2 items, 3 items, etc. It repeatedly scans the database to count the support of each pattern. On the other hand, pattern growth algorithms utilize a depth-first search instead of a breadth-first search, which only considers patterns in the database. Fuzzy frequent itemsets mining is also divided based on two type membership functions: the type-1 membership function and the type-2 membership function. The type-2 Fuzzy Set might deliver more reliable and agile decision-making by incorporating many uncertainty possibilities and more complex interactions between variables.

## 3.2 Type-1 membership function-based approach

In 1997, Chan et al. [33] presented the F-APACS algorithm for mining fuzzy association rules. They initiated the process by transforming quantitative attribute values into linguistic expressions and subsequently employed a modified variance examination to discover exciting correlations between attributes. In 1998, Kuok and colleagues [34] proposed a method for fuzzy mining that aimed to extract fuzzy association rules from numerical data stored in databases. Subsequently, in 1999, Hong and collaborators [12] introduced a fuzzy mining technique designed explicitly for extracting fuzzy rules from quantitative transaction data. In 2004, they further enhanced the approach by incorporating a GDF [35] strategy to reduce computational costs associated with deriving fuzzy frequent itemsets. The author presented a novel mining method to extract common patterns for building itemsets from quantitative databases using the Apriori Tid data structure. In 2003, Delgado et al. [36] tried to find a method to achieve the fuzzy association rules applicable to quantitative data and relational databases.

Because fuzzy-set processing from the tree structure is substantially more complex than crisp-set processing, fuzzy data mining based on pattern growth is uncommon. Several approaches for fuzzy data mining from tree structures have been presented. In 2005, Papadimitriou and Mavroudi [37] presented the fuzzy frequent-pattern tree (FFPT) technique for finding fuzzy association rules. In 2008, Hong et al. [38] used an FP-tree structure called FUFP-tree to reduce the execution time when new data is inserted or arrives. In 2010, the same FP-tree-like structure, referred to as the FFP-tree (fuzzy frequent pattern) [39], was employed by Lin et al. to find FFIs in quantitative databases. There are several limitations, but they are addressed in [40] [41]. In 2010, Lin et al. introduced a structured representation known as the Compressed Fuzzy Frequent Pattern (CFFP-tree) [40], and later, in 2014, they utilized the Upper Bound Fuzzy Frequent Pattern (UBFFP-tree) [41] structure for the discovery of FFIs. These structures, namely the CFFP-tree and UBFFP-tree, adopt a global sorting approach to reduce the count of tree nodes. In 2011, Mishra and Satapathy devised a strategy for mining frequent patterns in a fuzzified gene expression dataset [42]. It demonstrated that the fuzzy vertical dataset format produced more fuzzy frequent itemsets than the original. 2011 Mahmoudi et al. developed the ant colony system and multiple-level taxonomy [43]. In 2012, Chen and colleagues devised a fusion model utilizing the cumulative probability distribution approach to enhance multi-level fuzzy association rules [44]. He uses a combination of the cumulative probability distribution technique and multi-level taxonomy to generate multi-level fuzzy association rules level by level. In 2012, Chang and colleagues [45] introduced the Integrated Sequential Pattern Mining with Fuzzy Time Intervals (ISPFTI) algorithm, which is designed for extracting frequent sequential patterns from sequence databases. The approach incorporates fuzzy theory to analyze the time intervals between these frequent sequences. Additionally, in 2012, Watanabe and Fujioka explored the equivalence redundancy of fuzzy items and presented relevant theorems for fuzzy data mining [46]. Addressing computational challenges associated with fuzzy itemset mining, Hong and collaborators proposed a data-reduction strategy in 2013 [47]. In [48], Hong et al. suggested using an MFFP-tree structure and mining MFFP growth to find MFFIs (2014). Similar to this, Lin et al. created the CMFFP-tree [49] (2015) and UBMFFP-tree [50] (2015) ways to create MFFIs based on the CFFP-tree [40] and the UBFFP-tree [41], respectively. In [51], Lin et al. developed a fuzzy frequent itemsets (FFI)-Miner algorithm to mine the complete set of FFIs without candidate generation (2015). In this approach, consider the maximum cardinality mechanism to generate FFIs. In [13], Lin et al. developed an MFFI-miner technique and a fuzzy-list structure to find MFFIs (2017). To decrease the search space, shorten the running time, and shrink the running space, the author of this method employed two pruning strategies. In [52], Zhang et al. suggested the FC-Tree structure and FCFI-miner (Fuzzy closed frequent itemsets miner) for the objective of finding FFIs (2018). Several algorithms drawing from the foundation of fuzzy-set theory for discovering the required information were developed in progress [53][54] and [55] for different applications and domains. For the medical domain in paper [56], a novel fuzzy methodology, IFFP (Improved Fuzzy Frequent Pattern Mining), has been introduced by Dhanaseelan et al., which is based on fuzzy association rule mining for biological knowledge extraction (2021). The approach comprises two stages. During the first phase, fuzzy frequent itemsets are mined using the proposed algorithm IFFP. In the second phase, fuzzy association rules (ARs) are formed, which indicate whether the person belongs to the benign category (not dangerous) or malignant category (dangerous to health).

In the paper [34], the main focus is on the mining of fuzzy association rules from databases. Association rule mining is a fundamental data mining task that aims to discover exciting relationships or patterns in large datasets. Fuzzy association rules extend this concept by including uncertainty or imprecision in the discovered rules. The paper presents a concept called fuzzy association rules and introduces an algorithm designed for their identification. The authors propose a two-step process involving generating candidate rules and selecting the most interesting ones. The paper's novelty lies in its adaptation of fuzzy logic to association rule mining, which enables it to handle imprecise and uncertain data effectively.

In the paper [12], the authors suggest a dual-phase approach for extracting association rules from quantitative data. In the first step, they discretize the quantitative attributes to convert them into categorical attributes. This is achieved using Equal Width, Equal Frequency, or Clustering. In the second step, the Apriori algorithm is adapted to mine association rules from the discretized data. The paper presents experimental results, demonstrating the effectiveness of their approach on various datasets.

In the paper [36], the authors present a comprehensive model for fuzzy association rules, extending the conventional binary association rule framework to accommodate uncertainty and fuzziness in data. They discuss the theoretical foundations of fuzzy association rules, including representing fuzzy sets and calculating membership degrees. The paper also delves into the efficient mining algorithms for discovering fuzzy association rules from large datasets.

In paper [35], an essential contribution is incorporating fuzzy logic into the well-known Apriori algorithm, which allows for a more flexible and nuanced representation of itemsets relationships in data. The algorithm can handle uncertain or imprecise data using fuzzy logic, making it suitable for real-world applications where data may not be crisp. Moreover, the authors emphasize reducing computational time as a significant advantage of their approach. This is a critical consideration in data mining, as the analysis of large datasets can be computationally intensive.

The paper [37] proposed an approach to create fuzzy association rules utilizing FP trees. In cases where the fuzzy value of a fuzzy region within a transaction falls below the minimum support threshold, the region is excluded. Only the local fuzzy frequent 1-itemsets in each transaction are mined in this approach. Fuzzy patterns can be expressed directly without requiring fuzzy operations to create the required rules. This process complicates the interpretation of the mined fuzzy rules.

The paper [38] proposes an innovative approach to incrementally update frequent pattern trees, allowing faster and more efficient updates than traditional methods. The critical contributions of the paper include a new structure for representing frequent pattern trees and an incremental update algorithm that significantly reduces the computational cost of updating these trees as new data becomes available.

In the paper [40], the authors propose a compressed fuzzy FP-tree (CFFP-tree), a method that leverages the power of decision trees to mine fuzzy data efficiently. Decision trees are a popular tool in data mining, and by incorporating fuzzy logic into their construction and analysis, the authors aim to improve the handling of uncertain or imprecise data. They introduce a systematic framework for creating fuzzy decision trees, which allows for better representation and interpretation of data with inherent vagueness or uncertainty. These algorithms manage the linguistic term with the highest membership value among the transformed linguistic terms associated with an item.

In the paper [41], the authors introduce the concept of UBFFP trees, a data structure that efficiently stores upper-bound information to accelerate the mining process. They propose algorithms and techniques for constructing and manipulating UBFFP trees to improve the efficiency and scalability of fuzzy frequent itemsets mining.

In the paper [42], the authors proposed FP mining to handle a dataset of fuzzified gene expression. It demonstrated that many fuzzy frequent itemsets could be derived from the fuzzy vertical dataset format rather than the original one.

In the paper [43], the authors integrated a multi-level taxonomy, the ant colony system, and the fuzzy-set concept to generate multi-level fuzzy association rules from quantitative transactions. The algorithm they introduced comprised three main stages: calculating the minimum supports of items within a database, establishing multiple minimum supports for items, and extracting membership functions using the ACS algorithm.

In the paper [44], the authors suggested a fusion model that utilizes the cumulative probability distribution approach to enhance the multi-level fuzzy association rules. Through a step-by-step process, they generate multi-level fuzzy association rules by incorporating both multi-level taxonomy and the cumulative probability distribution approach.

In a paper [45], Chang et al. employed fuzzy theory to explore time intervals between frequent sequences. They created the Integrated Sequential Pattern Mining with Fuzzy Time Intervals (ISPFTI) algorithm to extract frequent sequential patterns from sequence databases. Initially, the algorithm derives fuzzy frequent sequential patterns with minimal fuzzy support. Subsequently, fuzzy frequent time sequential patterns are identified based on the fuzzy support of each time cluster.

In the paper [46], Watanabe and Fujioka have articulated the concept of equivalence redundancy in fuzzy items and introduced corresponding theorems in the context of fuzzy data mining. They have presented an Apriori-like methodology that leverages the equivalence redundancy inherent in fuzzy items, drawing inspiration from the redundancy principles associated with fuzzy association rules.

In the paper [47], the authors introduce a data-reduction strategy that effectively deals with the computational challenges of fuzzy itemsets mining. The proposed strategy gradually reduces the dataset in a way that balances the need for computational efficiency with maintaining the quality of the mined fuzzy itemsets.

The critical contribution of the paper [48] lies in its introduction of the MFFP tree, which serves as an efficient data structure for fuzzy mining tasks. The authors show how useful it is for identifying linguistic frequent itemsets, that is, collections of items that appear repeatedly in a dataset. Instead of extracting representative linguistic terms from a series of quantitative transactions, Hong et al. expand the fuzzy FP-tree into multiple fuzzy-term FP (MFFP) trees. This modification enables them to discover all fuzzy frequent itemsets.

In the paper [49], The authors tackle the issue of identifying frequent itemsets in fuzzy transaction data, where items exhibit varying degrees of membership in transactions. The paper introduces the Compressed Multiple Fuzzy-Term FP-tree (CMFFP-tree) algorithm to mine these fuzzy frequent itemsets efficiently. The algorithm enhances the standard FP-tree by incorporating support for fuzzy data, ensuring that the output includes comprehensive and correlated fuzzy frequent itemsets, irrespective of whether they originate from the same item. The CMFFP-tree structure stores the complete set of linguistic terms to facilitate mining fully fuzzy frequent itemsets. The algorithm retains the linguistic term with the highest membership value and includes other frequent linguistic terms to achieve this. Construction of the CMFFP-tree involves sorting multiple frequent linguistic terms in descending order based on their occurrence frequencies. Inherited from the CFFP tree [40], the CMFFP tree maintains linguistic terms in descending order of occurrence frequency, building the tree structure tuple by tuple through updated transactions. Each tree node is associated with an array that utilizes the minimum operation to store membership values of itemsets containing their super-items in the path. The CMFFP-mine algorithm, rooted in the CMFFP tree, streamlines the recursive process of the FP-growth-like approach, efficiently extracting related fuzzy frequent itemsets from the tree's paths.

In the paper [50], the authors suggested mining MFFIs using an upper-bound multiple fuzzy FP-tree (UBMFFP) approach in contrast to maximal cardinality ones used in UBFFP. The UBMFFP tree, as suggested, is adept at managing quantitative databases to extract multiple association rules, aiding decision-makers in arriving at more understandable conclusions [41]. The MFFP-tree [48] and the CMFFP-tree [49] approach generate more tree nodes than the UBMFFP-tree method concerning space complexity. The suggested UBMFFP tree employs a two-phase strategy to minimize the candidate count and accelerate execution times, aiming to lower overall time complexity.

In a paper [51], Lin et al. developed the Fuzzy Frequent Itemsets (FFI)-Miner algorithm to extract complete FFIs efficiently, eliminating the necessity for candidate generation. Utilizing a novel fuzzy-list structure, this algorithm retains essential data to support subsequent mining operations. An effective pruning strategy is also developed to reduce the search area and streamline the exploration process of locating the FFIs directly. This method generates FFIs by taking into account the maximum cardinality mechanism. The maximum cardinality mechanism can significantly decrease the computations required to find FFIs, but some information might be lost.

In paper [13], the algorithm known as MFFI-Miner was developed by Lin and colleagues to extract the complete set of multiple fuzzy frequent itemsets (MFFI) without relying on candidate generation. The fuzzy-list structure is used to store the data that is necessary for the subsequent mining step. This approach accelerates the mining procedure for identifying MFFIs by narrowing the search space and applying two efficient pruning techniques. A detail of this approach is shown in section 3.2.1, section.

The author of [52] suggested the FC-Tree structure and FCFI-miner (Fuzzy closed frequent itemsets miner) to find FFIS. In this method, the author used a superset pruning mechanism to speed up mining. Authors come upon MFFIs, which offer thorough details on all linguistic expressions in the fuzzy set.

The key focus of this paper [55] is creating an algorithm for mining incremental fuzzy association rules, which allows for the continuous updating of association rules as new data becomes available. This is particularly useful in dynamic data environments where the dataset evolves. The authors demonstrate the effectiveness of their approach by applying it to classification and regression tasks. By incorporating fuzzy logic into the association rule mining process, the model can handle imprecise and uncertain data, making it suitable for real-world applications where data quality may vary. Overall, the paper offers a valuable contribution to data mining by introducing an incremental approach that enhances the adaptability and accuracy of association rule mining in the context of classification and regression.

### 3.2.1 MFFI-miner approach [13]

Paper Title: "Efficient Mining of Multiple Fuzzy Frequent Itemsets ".

-Method: MFFI-Miner

-Data Structure: fuzzy-list structure, enumeration tree

-First transpose the quantitative dataset into a fuzzy dataset with fuzzy value.

For example, the quantitative database shown in Table 3.1 transposes into a fuzzy set say D' shown in Table 3.2, and the predefined membership functions shown in Figure 3.2 are used to convert the quantitative value of each item into several fuzzy linguistic terms (fuzzy itemsets) with their membership degrees.

Table 3.1: Quantitative database

| TID | Items and their quantities |
|---|---|
| 1 | C:3, D:2, E:1 |
| 2 | B:1, C:2, D:1 |
| 3 | B:3, C:3, E:1 |
| 4 | A:3, C:5, D:3 |
| 5 | A:1, B:1, C:2, D:1 |
| 6 | B:1, D:1, E:2 |
| 7 | A:4, B:3, D:5, E:3 |
| 8 | B:1, C:2, D:1 |

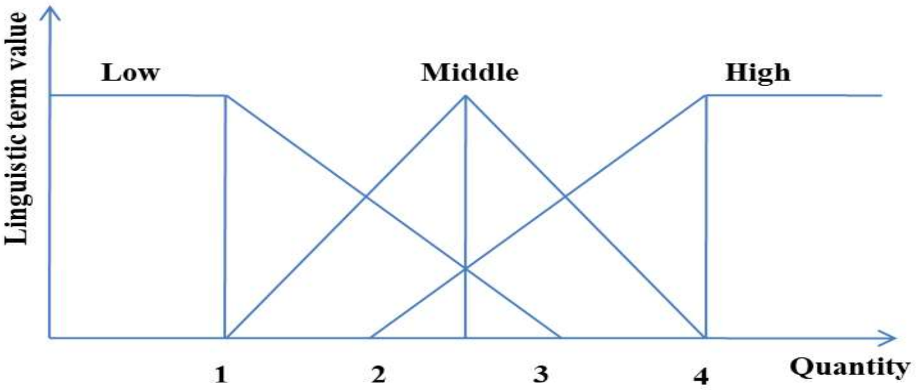


Fig: 3.2: Membership function

Table 3.2: Transpose fuzzy dataset

| TID | Transformed Linguistic terms |
|---|---|
| 1 | 0.67/C.M + 0.5/C.H, 0.5/D.L+0.67/D.M, 1/E.L |
| 2 | 1/B.L, 0.5/C.L+0.67/C.M, 1/D.L |
| 3 | 0.67/B.M + 0.5/B.H, 0.67/C.M + 0.5/C.H, 1/E.L |
| 4 | 0.67/A.M + 0.5/A.H, 1/C.H, 0.67/D.M + 0.5/D.H |
| 5 | 1/A.L, 1/B.L, 0.5/C.L+0.67/C.M, 1/D.L |
| 6 | 1/B.L, 1/D.L, 0.5/E.L+0.67/E.M |
| 7 | 1/A.H, 0.67/B.M + 0.5/B.H, 1/D.H, 0.67/E.M + 0.5/E.H |
| 8 | 1/B.L, 0.5/C.L+0.67/C.M, 1/D.L |

-Next Step: Prune items that do not satisfy minimum support threshold and generate frequent 1-itemset ($L_1$) from fuzzy dataset D’.

For generating $L_1$, find the support count of each linguistic term and discard linguistic terms that do not satisfy the min support threshold. The fuzzy values of the same fuzzy itemsets are summed up together as the support value of the fuzzy Itemsets. The support count of each linguistic term is shown in Table 3.3. For example, consider min support $\delta$=2.0 here, then $L_1$ is shown below in Table 3.4.

Table 3.3: Support counts of linguistic terms

| Linguistic Term | Support Count |
|---|---|
| AL | 1 |
| AM | 0.67 |
| AH | 1.5 |
| BL | 4 |
| BM | 1.34 |
| BH | 1 |
| CL | 1.5 |
| CM | 3.35 |
| CH | 2 |
| DL | 4.5 |
| DM | 1.34 |
| DH | 1.5 |
| EL | 2.5 |
| EM | 1.34 |
| EH | 0.5 |

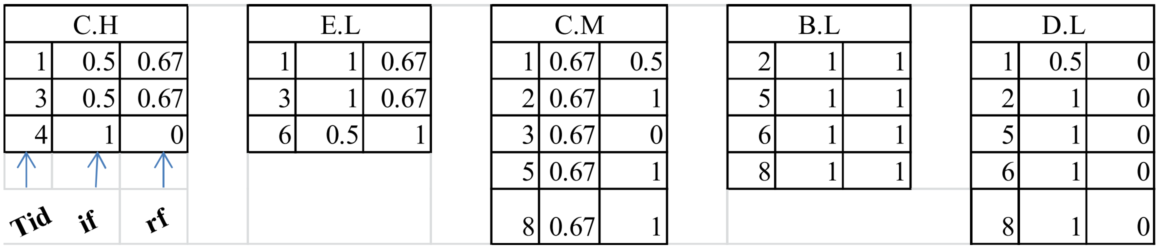
Table 3.4: L1 (fuzzy 1-frequent itemset)

| Linguistic Term | Support Count |
|---|---|
| BL | 4 |
| CM | 3.35 |
| CH | 2 |
| DL | 4.5 |
| EL | 2.5 |

-Next Step: Fuzzy-list Construction

L1 is used to build a fuzzy-list structure for keeping the necessary information. Each transaction in D' is sorted in ascending order according to the support count of linguistic terms; for example, the sorted order is CH, EL, CM, BL, and DL. The fuzzy-list contains *Tid* (Transaction Id), *if* (Internal fuzzy value), and *rf* (Resting fuzzy value), shown in Figure 3.3.

The fuzzy value of linguistic terms in a transaction is defined as the internal fuzzy value and denoted as *if*. The resting fuzzy value is calculated by performing the maximum operation to get the maximum fuzzy value among the resting linguistic terms in the transaction, denoted as *rf*.



| C.H | | |
|---|---|---|
| 1 | 0.5 | 0.67 |
| 3 | 0.5 | 0.67 |
| 4 | 1 | 0 |

| E.L | | |
|---|---|---|
| 1 | 1 | 0.67 |
| 3 | 1 | 0.67 |
| 6 | 0.5 | 1 |

| C.M | | |
|---|---|---|
| 1 | 0.67 | 0.5 |
| 2 | 0.67 | 1 |
| 3 | 0.67 | 0 |
| 5 | 0.67 | 1 |
| 8 | 0.67 | 1 |

| B.L | | |
|---|---|---|
| 2 | 1 | 1 |
| 5 | 1 | 1 |
| 6 | 1 | 1 |
| 8 | 1 | 1 |

| D.L | | |
|---|---|---|
| 1 | 0.5 | 0 |
| 2 | 1 | 0 |
| 5 | 1 | 0 |
| 6 | 1 | 0 |
| 8 | 1 | 0 |

Fig 3.3: Initial Constructed fuzzy-list

The authors use an enumeration tree-like representation for searching MFFIs from generated fuzzy nodes (fuzzy-list). The authors used the depth-first search strategy to traverse the enumeration tree and decide whether the child (superset) nodes must be generated and explored. They use two pruning strategies to minimize search space. Strategy 1: For a fuzzy node (itemsets), if its internal fuzzy value (*if*) is less than the minimum support threshold, it is not considered an FFI, and the supersets of that are not generated. Strategy 2: For a fuzzy node (itemsets), if its resting fuzzy value (*rf*) is less than the minimum support threshold, then the supersets of that are not generated.

For the fuzzy-list structures of fuzzy k-itemsets ($k \geq 2$), it is unnecessary to scan the revised database (D') and perform the intersection operation of the fuzzy-list by the tids in k fuzzy-list structures. For example, the 'C.H' fuzzy-node internal fuzzy value (*if*) satisfies the min support threshold ($\delta$), but the resting fuzzy value (*rf*) does not fulfill the min support threshold ($\delta$), so according to strategy two supersets of 'C.H' fuzzy node not generated. The remaining node satisfies the criteria, according to the example generated 2-fuzzy-list (nodes) shown in Figure 3.4. Here, six nodes are generated, but only three nodes are FFIs, so it is unnecessary to join the operation to generate more non-frequent fuzzy nodes. Apply the same procedure to generate the subsequent fuzzy-list nodes, as per the example generated 3-fuzzy-list node shown in Figure 3.5.

| E.L-C.M | | |
|---|---|---|
| 1 | 0.67 | 0.5 |
| 3 | 0.67 | 0 |

| E.L-B.L | | |
|---|---|---|
| 6 | 0.5 | 1 |

| B.L-D.L | | |
|---|---|---|
| 2 | 1 | 0 |
| 5 | 1 | 0 |
| 6 | 1 | 0 |
| 8 | 1 | 0 |

| E.L-D.L | | |
|---|---|---|
| 1 | 0.5 | 0 |
| 6 | 0.5 | 0 |

| C.M-B.L | | |
|---|---|---|
| 2 | 0.67 | 1 |
| 5 | 0.67 | 1 |
| 8 | 0.67 | 1 |

| C.M-D.L | | |
|---|---|---|
| 1 | 0.5 | 0 |
| 2 | 0.67 | 0 |
| 5 | 0.67 | 0 |
| 8 | 0.67 | 0 |

Fig. 3.4: Constructed 2-fuzzy-list

| [illegible] | | |
|---|---|---|
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] |

Fig.3.5: Constructed 3-fuzzy-list

As an example, the representation of an enumeration tree for searching MFFIs from generated fuzzy nodes is shown in Figure 3.6.

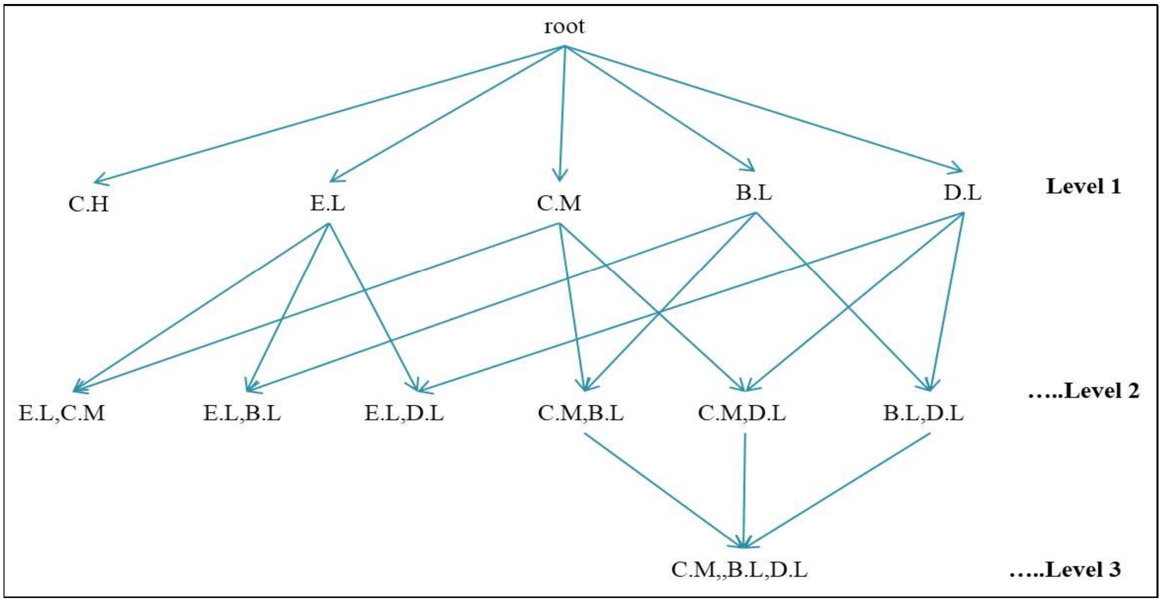


Fig. 3.6: enumeration tree

Summary: The authors proposed two efficient pruning strategies to minimize node join operation and improve running time. However, this approach still generates more non-frequent fuzzy nodes, which require unnecessary join operations.

## 3.3 Type-2 membership function-based approach

The solution mentioned above solely counters type-1 fuzzy-set theory, which ignores uncertainty. The fuzzy-set idea with type-2 membership function by Mendel and John [57] (2002), Castillo and Melin [58] (2008), and Hagras [59] (2008) was then put out and improved to more effectively present the acquired information with uncertainty. To merge pattern mining and type-2 fuzzy sets, in 2015, Chen et al. [60] applied the standard level-wise like-Apriori method for mining level-wise fuzzy type-2 frequent patterns. However, the procedure necessitates generating large numbers of candidates, which is ineffective for the mining task. To store the information for the mining process, a list-based approach proposed by Lin et al., the strategy still needs to investigate many candidates for determining the true FFIs due to inefficient pruning algorithm and loose upper bound value on the pattern, which are not frequent. Following this, Lin and colleagues (2016) introduced a method based on lists to retain essential information in identifying frequent items. This approach still needs to investigate numerous candidates to get the actual fuzzy frequent itemsets because it requires more effective search space trimming strategies and a flexible upper bound measure on the patterns that could be more promising. Lin et al. [14] employed a complex fuzzy list (CFL)-structure to find MFFIs (2020), and that was similar to the fuzzy list structure from [13].

In paper [57], Mendel and John provide a comprehensive and accessible introduction to Type-2 fuzzy sets, a more advanced and powerful extension of Type-1 fuzzy sets. They present a clear and concise overview of the theoretical foundations of Type-2 fuzzy sets, discussing the concept of uncertainty in fuzzy sets and how Type-2 fuzzy sets can model higher degrees of uncertainty. The authors also discuss practical aspects of Type-2 fuzzy sets, including their representation, operations, and applications.

In papers [58] and [59], the authors comprehensively overview Type-2 fuzzy logic controllers, including their theoretical foundations and practical applications. It highlights the benefits of employing Type-2 fuzzy logic in handling uncertain and dynamic situations, making it a valuable tool for real-world control systems.

In their publication [60], Chen et al. introduced a methodology for integrating type-2 fuzzy sets into pattern mining. They adapted the traditional level-wise (similar to Apriori) method to extract fuzzy type-2 frequent patterns at different levels systematically. This method, however, involves producing a considerable number of candidates with a high level of time complexity, making it inefficient for the mining task. Additionally, it employs the maximal scalar cardinality method to extract just one linguistic term per item. As a result, the information obtained may lack comprehensiveness, potentially leading to an insufficient knowledge base for decision-making purposes.

In their publication [61], Lin and colleagues introduced a method based on lists to extract type-2 fuzzy frequent patterns effectively. This approach aims to retain comprehensive information during the mining process, improving mining performance when contrasted with the level-wise method. Even with effective pruning techniques and a lenient upper limit on unfavorable patterns, this method must evaluate numerous candidates to extract the fuzzy frequent patterns.

The paper [14] focuses on efficiently extracting multiple fuzzy frequent patterns from a database. This is achieved by incorporating membership functions derived from type-2 fuzzy-set theory. Additionally, a streamlined structure is devised to retain comprehensive information, and the implementation of effective pruning strategies is essential to diminish the search space size. This, in turn, enhances the overall performance of the pattern mining process. Details of this paper are shown in section 3.3.1.

### 3.3.1 EFM approach [14]

Paper Title: “Mining Multiple Fuzzy Frequent Patterns with Compressed List Structures“
-Method: EFM
-Data Structure: fuzzy-list structure, enumeration tree
-First, transpose the quantitative dataset into a fuzzy dataset with fuzzy values using the type-2 membership function.
For instance, the numeric data presented in Table 3.1 transforms into a fuzzy set, as illustrated in Table 3.5. The predefined type-2 membership functions depicted in Figure 3.7 are then employed to convert the numerical value associated with each item into

multiple fuzzy linguistic terms (referred to as fuzzy itemsets) along with their corresponding membership degrees. Next, the author used the centroid type reduction approach to generate the final fuzzy dataset, say D' shown in Table 3.6. The centroid type-reduction method uses the average of lower and upper intervals.

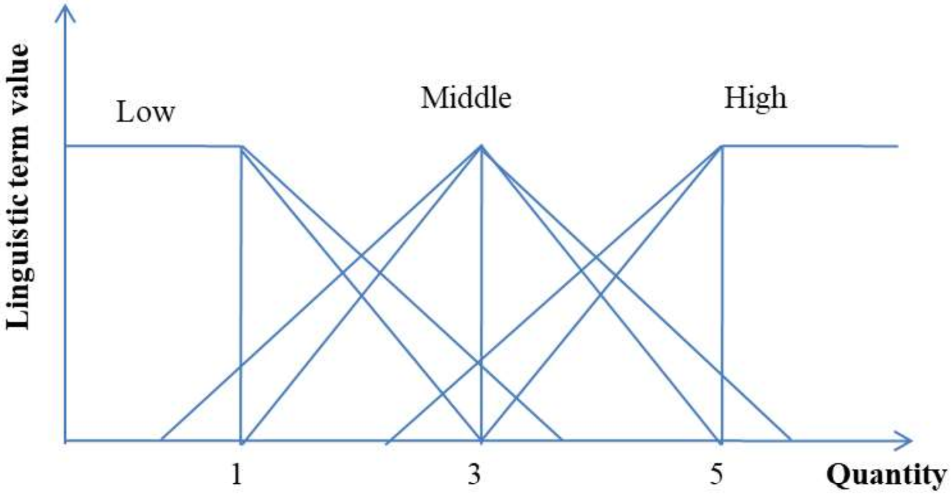


Fig. 3.7: predefined type-2 membership function

Table 3.5: Transpose fuzzy dataset using type-2 function

| Tid | Transform Fuzzy Dataset |
|---|---|
| 1 | 0,0.25/C.L + 1,1/C.M + 0,0.25/C.H , 0.5,0.62/D.L + 0.5,0.62/D.M , 1,1/E.L + 0,0.25/E.M |
| 2 | 1,1/B.L + 0,0.25/B.M , 0.5,0.62/C.L + 0.5,0.62/C.M , 1,1/D.L + 0,0.25/D.M |
| 3 | 0,0.25/B.L + 1,1/B.M + 0,0.25/B.H , 0,0.25/C.L + 1,1/C.M + 0,0.25/C.H , 1,1/E.L + 0,0.25/E.M |
| 4 | 0,0.25/A.L + 1,1/A.M + 0,0.25/A.H , 0,0.25/C.M + 1,1/C.H , 0,0.25/D.L + 1,1/D.M + 0,0.25/D.H |
| 5 | 1,1/A.L + 0,0.25/A.M , 1,1/B.L + 0,0.25/B.M , 0.5,0.62/C.L + 0.5,0.62/C.M , 1,1/D.L + 0,0.25/D.M |
| 6 | 1,1/B.L + 0,0.25/B.M , 1,1/D.L + 0,0.25/D.M , 0.5,0.62/E.L + 0.5,0.62/E.M |
| 7 | 0.5,0.62/A.M + 0.5,0.62/A.H , 0,0.25/B.L + 1,1/B.M + 0,0.25/B.H , 0,0.25/D.M + 1,1/D.H , 0,0.25/E.L + 1,1/E.M + 0,0.25/E.H |

Table 3.6: Final fuzzy dataset by centroid reduction approach

| Tid | Final Fuzzy data |
|---|---|
| 1 | 0.13/C.L + 1/C.M + 0.13/C.H, 0.56/D.L + 0.56/D.M, 1/E.L + 0.13/E.M |
| 2 | 1/B.L + 0.13/B.M, 0.56/C.L + 0.56/C.M, 1/D.L + 0.13/D.M |
| 3 | 0.13/B.L + 1/B.M + 0.13/B.H, 0.13/C.L + 1/C.M + 0.13/C.H, 1/E.L + 0.13/E.M |
| 4 | 0.13/A.L + 1/A.M + 0.13/A.H, 0.13/C.M + 1/C.H, 0.13/D.L + 1/D.M + 0.13/D.H |
| 5 | 1/A.L + 0.13/A.M, 1/B.L + 0.13/B.M, 0.56/C.L + 0.56/C.M, 1/D.L + 0.13/D.M |
| 6 | 1/B.L + 0.13/B.M, 1/D.L + 0.13/D.M, 0.56/E.L + 0.56/E.M |
| 7 | 0.56/A.M + 0.56/A.H, 0.13/B.L + 1/B.M + 0.13/B.H, 0.13/D.M + 1/D.H, 0.13/E.L + 1/E.M + 0.13/E.H |

Next, find the scalar cardinality of each linguistic term. To discover the complete information of MFFPs, the multiple linguistic terms of an itemset are considered in the derived knowledge. The following Table 3.7 shows the scalar cardinality (fuzzy value) of each linguistic term for running an example.

Table 3.7: Fuzzy value of each linguistic term

| Linguistic Term | Fuzzy Value |
|---|---|
| AL | 1.13 |
| AM | 1.69 |
| AH | 0.69 |
| BL | 4.26 |
| BM | 2.52 |
| BH | 0.26 |
| CL | 1.94 |
| CM | 3.81 |
| CH | 1.26 |
| DL | 4.69 |
| DM | 2.21 |
| DH | 1.13 |
| EL | 2.69 |
| EM | 1.82 |
| EH | 0.13 |

The author proposed the Efficient Fuzzy frequent Mining (EFM) method and compressed fuzzy-list (CFL) structure for mining MFFPs. The linguistic terms are sorted in ascending order to maintain the downward closure property for building the compressed fuzzy-list (CFL)-structure. For running example, Table 3.8 shows ascending order linguistic terms whose satisfy min support threshold δ=2.0 is considered an $L_1$.

Table 3.8: Ordered linguistic terms

| Linguistic Term | Fuzzy Value |
|---|---|
| DM | 2.21 |
| BM | 2.52 |
| EL | 2.69 |
| CM | 3.81 |
| BL | 4.26 |
| DL | 4.69 |

The compressed Fuzzy-list contains *Tid* (Transaction ID), *if* (Internal fuzzy value), and *rf* (Resting fuzzy value), shown below in Figure 3.8.

The fuzzy value of linguistic terms in a transaction is defined as the internal fuzzy value and denoted as *if*. The resting fuzzy value is calculated by performing the maximum operation to get the maximum fuzzy value among the resting linguistic terms in the transaction, denoted as *rf*.

| D.M | | |
|---|---|---|
| 1 | 0.56 | 1 |
| 2 | 0.13 | 1 |
| 4 | 1 | 0.13 |
| 5 | 0.13 | 1 |
| 6 | 0.13 | 1 |
| 7 | 0.13 | 1 |
| 8 | 0.13 | 1 |

| B.M | | |
|---|---|---|
| 2 | 0.13 | 1 |
| 3 | 1 | 1 |
| 5 | 0.13 | 1 |
| 6 | 0.13 | 1 |
| 7 | 1 | 0.13 |
| 8 | 0.13 | 1 |

| E.L | | |
|---|---|---|
| 1 | 1 | 1 |
| 3 | 1 | 0.13 |
| 6 | 0.56 | 1 |
| 7 | 0.13 | 0.13 |
| ↑ | ↑ | ↑ |
| Tid | if | rf |

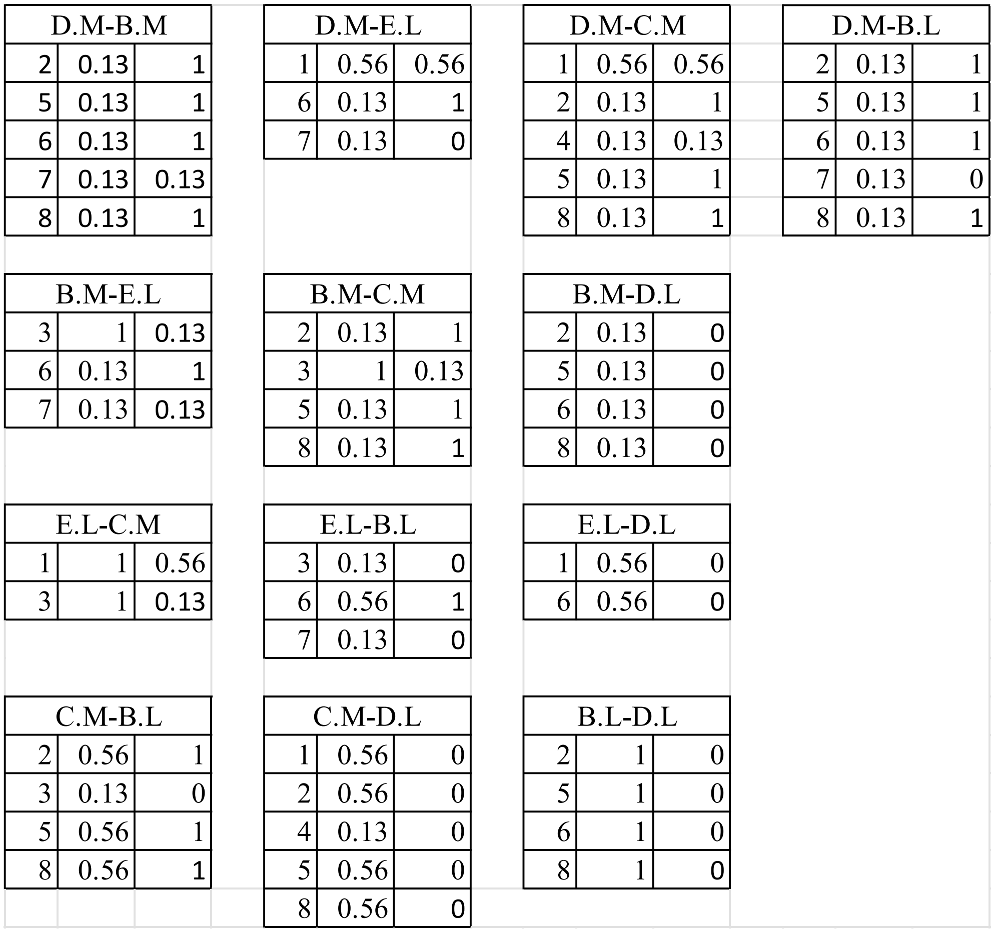
| C.M | | |
|---|---|---|
| 1 | 1 | 0.56 |
| 2 | 0.56 | 1 |
| 3 | 1 | 0.13 |
| 4 | 0.13 | 0.13 |
| 5 | 0.56 | 1 |
| 8 | 0.56 | 1 |

| B.L | | |
|---|---|---|
| 2 | 1 | 1 |
| 3 | 0.13 | 0 |
| 5 | 1 | 1 |
| 6 | 1 | 1 |
| 7 | 0.13 | 0 |
| 8 | 1 | 1 |

| D.L | | |
|---|---|---|
| 1 | 0.56 | 0 |
| 2 | 1 | 0 |
| 4 | 0.13 | 0 |
| 5 | 1 | 0 |
| 6 | 1 | 0 |
| 8 | 1 | 0 |

Fig 3.8: Compressed fuzzy-list

After CFLs-structures are generated, the EFM method uses two pruning strategies to reduce the space searching, which uses the Sup (*if*) and rSup (*rf*) of such a list X to decide whether to search the extension of X. First strategy is described as For an itemsets X, if Sup(X) is less than the minimum support threshold, then any supersets (extension) of X is not MFFPs and should be pruned. The second strategy is described as follows: For an itemsets X, if the relative remaining support rSup(X) is less than the minimum support threshold, then any supersets (extension) of X are not MFFPs and should be discarded.
In the running example, all nodes in L1 satisfy the min support threshold for both criteria Sup (if) and rSup (rf), so the join operation applies. In join use intersection operation on Tid, the resultant of 2-CFL is shown in Figure 3.9.



| D.M-B.M | | |
|---|---|---|
| 2 | 0.13 | 1 |
| 5 | 0.13 | 1 |
| 6 | 0.13 | 1 |
| 7 | 0.13 | 0.13 |
| 8 | 0.13 | 1 |

| D.M-E.L | | |
|---|---|---|
| 1 | 0.56 | 0.56 |
| 6 | 0.13 | 1 |
| 7 | 0.13 | 0 |

| D.M-C.M | | |
|---|---|---|
| 1 | 0.56 | 0.56 |
| 2 | 0.13 | 1 |
| 4 | 0.13 | 0.13 |
| 5 | 0.13 | 1 |
| 8 | 0.13 | 1 |

| D.M-B.L | | |
|---|---|---|
| 2 | 0.13 | 1 |
| 5 | 0.13 | 1 |
| 6 | 0.13 | 1 |
| 7 | 0.13 | 0 |
| 8 | 0.13 | 1 |

| B.M-E.L | | |
|---|---|---|
| 3 | 1 | 0.13 |
| 6 | 0.13 | 1 |
| 7 | 0.13 | 0.13 |

| B.M-C.M | | |
|---|---|---|
| 2 | 0.13 | 1 |
| 3 | 1 | 0.13 |
| 5 | 0.13 | 1 |
| 8 | 0.13 | 1 |

| B.M-D.L | | |
|---|---|---|
| 2 | 0.13 | 0 |
| 5 | 0.13 | 0 |
| 6 | 0.13 | 0 |
| 8 | 0.13 | 0 |

| E.L-C.M | | |
|---|---|---|
| 1 | 1 | 0.56 |
| 3 | 1 | 0.13 |

| E.L-B.L | | |
|---|---|---|
| 3 | 0.13 | 0 |
| 6 | 0.56 | 1 |
| 7 | 0.13 | 0 |

| E.L-D.L | | |
|---|---|---|
| 1 | 0.56 | 0 |
| 6 | 0.56 | 0 |

| C.M-B.L | | |
|---|---|---|
| 2 | 0.56 | 1 |
| 3 | 0.13 | 0 |
| 5 | 0.56 | 1 |
| 8 | 0.56 | 1 |

| C.M-D.L | | |
|---|---|---|
| 1 | 0.56 | 0 |
| 2 | 0.56 | 0 |
| 4 | 0.13 | 0 |
| 5 | 0.56 | 0 |
| 8 | 0.56 | 0 |

| B.L-D.L | | |
|---|---|---|
| 2 | 1 | 0 |
| 5 | 1 | 0 |
| 6 | 1 | 0 |
| 8 | 1 | 0 |

Fig 3.9: 2-item CFL

In 2-CFL 'D.M-B.M', 'D.M-E.L', 'D.M-C.M', 'D.M-B.L', 'B.M-E.L', 'B.M-C.M', 'B.M-D.L', 'E.L-B.L', 'E.L-D.L' and ‘C.M-B.L’ nodes internal fuzzy value (*if*) not satisfy min support threshold δ so discarded that. The remaining 2-CFL nodes that satisfy the min support threshold δ are shown in Figure 3.10 and are considered 2-FFIs. In 2-FFIs, all nodes resting fuzzy value (*rf*) do not satisfy the min support threshold δ, so its superset is impossible.

| E.L-C.M | | |
|---|---|---|
| 1 | 1 | 0.56 |
| 3 | 1 | 0.13 |

| C.M-D.L | | |
|---|---|---|
| 1 | 0.56 | 0 |
| 2 | 0.56 | 0 |
| 4 | 0.13 | 0 |
| 5 | 0.56 | 0 |
| 8 | 0.56 | 0 |

| B.L-D.L | | |
|---|---|---|
| 2 | 1 | 0 |
| 5 | 1 | 0 |
| 6 | 1 | 0 |
| 8 | 1 | 0 |

Fig 3.10: 2-FFI

This method still generates more node (CFL) counts, requiring more execution time to search space exploration.

## 3.4 Summarization

In summary,

- Many scholars aim to enhance efficiency by minimizing the count of candidate itemsets and database scans.
- Existing methods are time-consuming to discover fuzzy frequent itemsets.
- Existing state-of-the-art methods scan the database more than once, ultimately increasing execution time.
- Most methods generate a very high number of node join counts (candidate itemsets) in fuzzy frequent itemsets mining, ultimately increasing execution time.
- The authors used two efficient pruning strategies discussed in the latest fuzzy frequent itemsets mining, MFFI-miner [13] and EFM [14]. These methods also unnecessarily generate massive node join counts due to increasing running time.

Therefore, it requires an efficient method that scans the database only once and generates fewer node join counts, ultimately improving execution time.

# CHAPTER 4

# An Adjacency matrix based Multiple Fuzzy Frequent Itemsets mining

The main contribution of this research is to design an efficient multiple fuzzy frequent itemsets mining method using an adjacency matrix. Proposed methods AMFFI scan database only once and reduce the number of node join counts (candidate itemsets) by pruning un-frequent itemsets extracted from the adjacency matrix.

The research methodology comprises developing a novel approach to mining MFFIs using an adjacency matrix and fuzzy-tid-list structures The AMFFI proposed model producing MFFIs in two steps as shown in Figure 4.1. First, construct an adjacency matrix and fuzzy-tid list from the quantitative dataset D in phase 1. The next step is using the AMFFI method to find MFFIs from the constructed adjacency matrix and fuzzy-tid-list. The suggested method efficiently creates full MFFIs by performing a single database scan. Section 4.1 discusses the Adjacency matrix and fuzzy-tid-list construction, and 4.2 discuss the AMFFI-miner to mine MFFIs.

## 4.1 Adjacency matrix and fuzzy-tid-list construction phase

During the first phase, the algorithm transforms the quantitative transaction values into a fuzzy set using the member function with several linguistic terms. In this approach, we use the type-1 membership function.

There are different types of type-1 membership functions according to linguistic terms such as 2-term, 3-term, 5-term, etc. Consider the membership function '£1' for Low, Middle, and High linguistic terms. Adjacency matrix AdjMat (M) of size (m*3) X (m*3) should first be constructed, where 'm' represents the overall count of items within the initial dataset D. In this case, for the matrix space total items required are three times to the original itemsets. The membership function determines the size of the matrix.

$$AdjMat\ (M) = (m * t)\ \times (m * t) \tag{4.1}$$

Where ‘m’ represents the overall count of items used in the original quantitative dataset, and variable 't' corresponds to the specific fuzzy region applied within the membership function of the t-term.

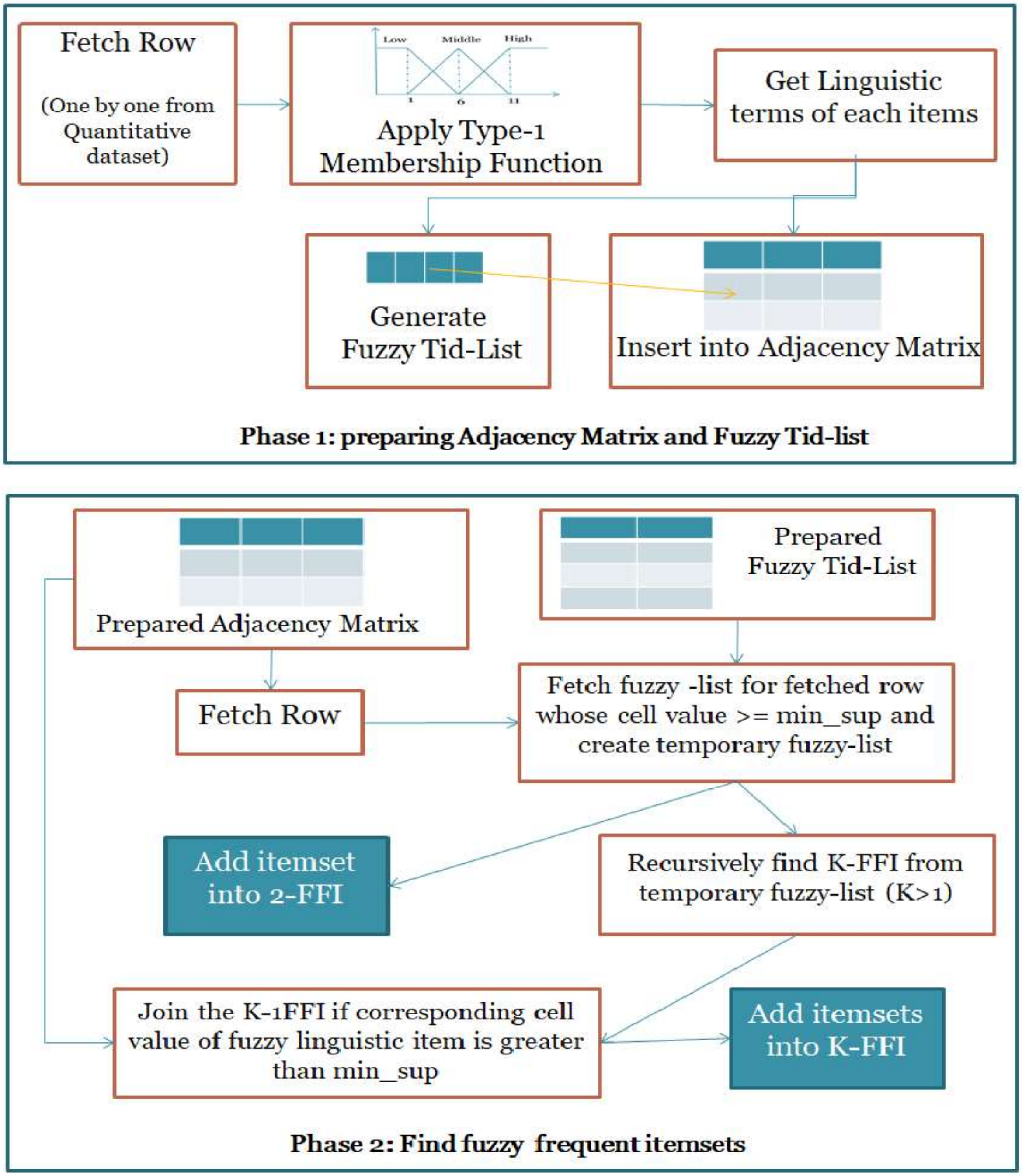

Fetch Row
(One by one from Quantitative dataset)
Low
Middle
High
1
6
11
Apply Type-1 Membership Function
Get Linguistic terms of each items
Generate Fuzzy Tid-List
Insert into Adjacency Matrix
Phase 1: preparing Adjacency Matrix and Fuzzy Tid-list
Prepared Adjacency Matrix
Prepared Fuzzy Tid-List
Fetch Row
Fetch fuzzy -list for fetched row whose cell value >= min_sup and create temporary fuzzy-list
Add itemset into 2-FFI
Recursively find K-FFI from temporary fuzzy-list (K>1)
Join the K-1FFI if corresponding cell value of fuzzy linguistic item is greater than min_sup
Add itemsets into K-FFI
Phase 2: Find fuzzy frequent itemsets

Fig 4.1: AMFFI Proposed Model

The quantitative dataset of the transaction $T_q$ with the TID q is converted into a fuzzy dataset and is achieved through applying the membership function £1.

Create a pair of converted fuzzy itemsets from transaction $T_q$ for various fuzzy variables. The correspondence cell value of the adjacency matrix should be updated by adding the minimal fuzzy value of each pair and entering it into the corresponding fuzzy-tid-list.

$$\text{AdjMat}(L_i, L_j) = \text{AdjMat}(L_i, L_j) + \min(f_{wiq}, f_{wjq}), \tag{4.2}$$

Li and Lj are fuzzy items whose fuzzy values are $f_{wiq}$ and $f_{wjq}$, respectively. If $L_i$ and $L_j$ don't have fuzzy-tid-lists, create them. The transaction id q (TID of $T_q$) and minimum fuzzy value of the pair as min ($f_{wiq}$, $f_{wjq}$) were added to this fuzzy-Tid-list.

The construction of the Adjacency matrix and fuzzy-tid-list is shown in Algorithm 4.1.

***Algorithm 4.1: Adjacency Matrix and Fuzzy-Tid-list construction***

```
Input: Quantitative dataset D, No. of Items M
Output: Adjacency matrix AM and Fuzzy-Tid-list FTL
Step 1: Initialize Matrix AM for (M * no of the fuzzy region) X (M * no of the fuzzy
region)

Step 2: Initialize TID=1
Step 3: Read line L from D
Step 4: Repeat through step 7 while L is not the end of file D
Step 5:Apply the membership function on each item's quantitative value in L and create
        fuzzy linguistic terms fl[ ] of all items.
Step 6: Store fuzzy value in adjacency matrix AM for all co-occurrences of fuzzy
        linguistic terms
          Define FV= min (fuzzy value of fl[i], fuzzy value of  fl[j])
         AM (fl[i], fl[j]) + =   FV
          Create Fuzzy-tid-list of “’ fl[i] ‘+’ fl[j]’ “if not exist
          Create element with (TID, FV) & insert into Fuzzy-tid-list of “ ’ fl[i] ‘+’ fl[j]’ “
Step 7: Increment TID by 1 and Read the Next Line from D into L
Step 8: Finished.
```

Let's take an instance of the numerical dataset D, as illustrated in Table 4.1, and the membership function £1, depicted in Figure 4.2 as an illustration. There are five items (A, B, C, D, E); accordingly, construct the adjacency matrix shown in Figure 4.3.

Table 4.1: Quantitative sample Dataset

| TID | Item with Quantity |
|---|---|
| 1 | A-4, B-3, C-2, D-2 |
| 2 | B-3, C-2, E-3 |
| 3 | A-5, B-3, C-4, E-4 |
| 4 | A-2, C-1, D-3 |
| 5 | A-4, B-2, C-5 |
| 6 | B-3, C-3, D-2, E-2 |
| 7 | C-3, E-2 |

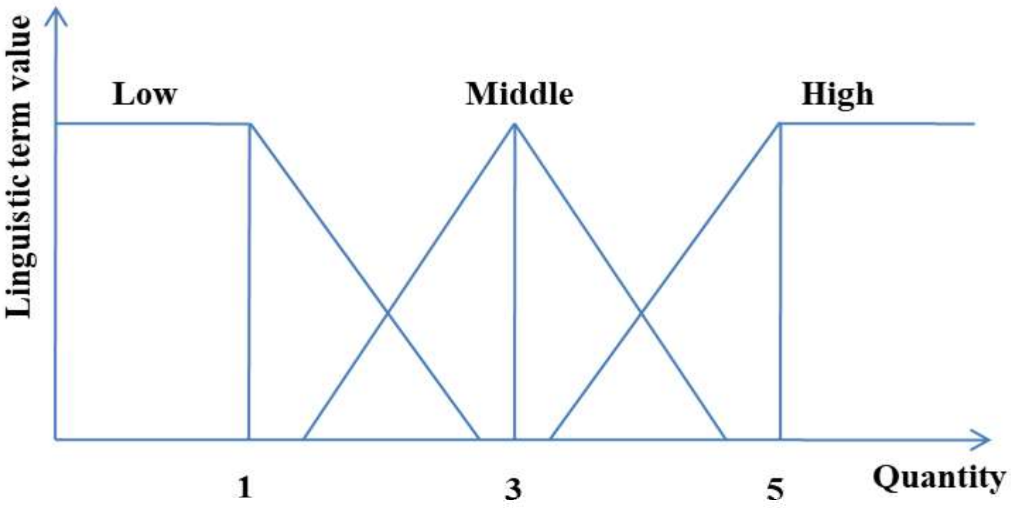
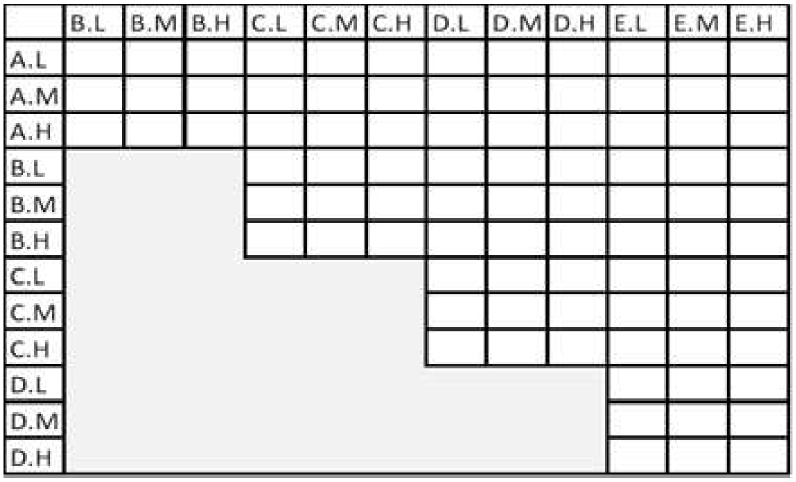


Fig.4.2: Type-1 Membership Function



| | B.L | B.M | B.H | C.L | C.M | C.H | D.L | D.M | D.H | E.L | E.M | E.H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A.L | | | | | | | | | | | | |
| A.M | | | | | | | | | | | | |
| A.H | | | | | | | | | | | | |
| B.L | ▨ | ▨ | ▨ | | | | | | | | | |
| B.M | ▨ | ▨ | ▨ | | | | | | | | | |
| B.H | ▨ | ▨ | ▨ | | | | | | | | | |
| C.L | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | | | | |
| C.M | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | | | | |
| C.H | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | | | | |
| D.L | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | |
| D.M | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | |
| D.H | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | |

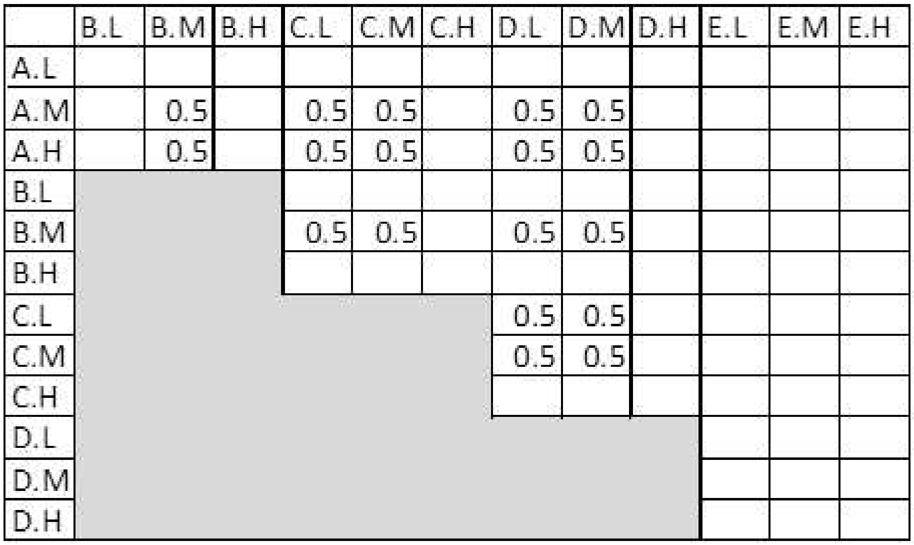
Fig. 4.3: Adjacency Matrix

Scan the first transaction from dataset D and apply the membership function £$_1$ to create a fuzzy set, as shown in Table 4.2.

Table 4.2: Fuzzy set for the first row

| TID | Items | Items Linguistic terms |
|---|---|---|
| 1 | A B C D::4 3 2 2 | 0.5/AM + 0.5/AH, 1/BM, 0.5/CL + 0.5/CM, 0.5/DL + 0.5/DM |

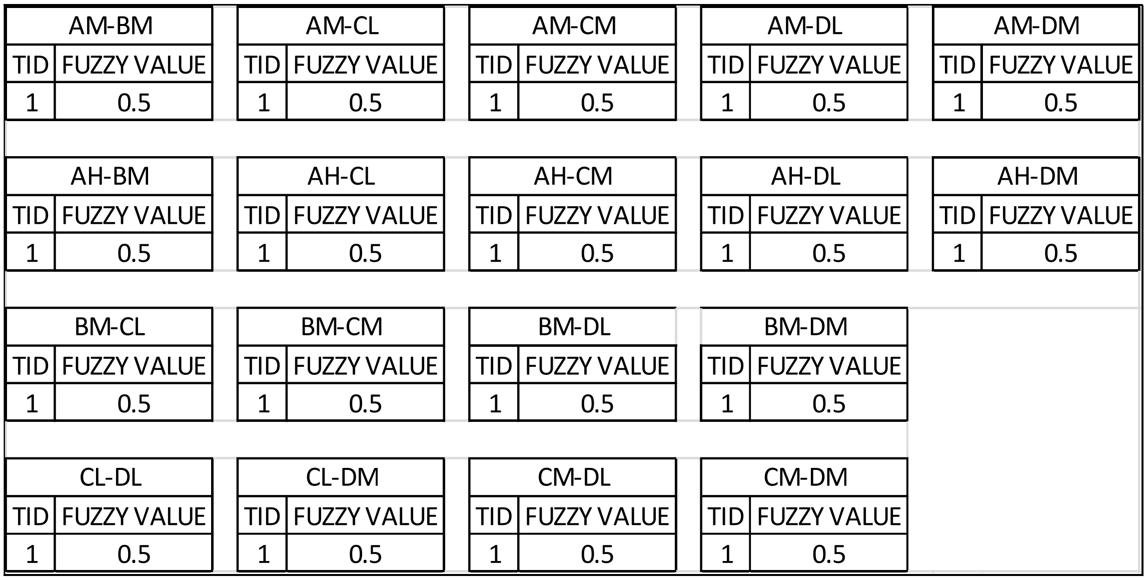
So created pair of transformed fuzzy itemsets are "AM-BM, AM-CL, AM-CM, AM-DL, AM-DM, AH-BM, AH-CL, AH-CM, AH-DL, AH-DM, BM-CL, BM-CM, BM-DL, BM-DM, CL-DL, CL-DM, CM-DL, and CM-DM". An adjacency matrix should be updated with a minimum fuzzy value for all pair co-occurrences, as shown in Figure 4.4.



| | B.L | B.M | B.H | C.L | C.M | C.H | D.L | D.M | D.H | E.L | E.M | E.H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A.L | | | | | | | | | | | | |
| A.M | | 0.5 | | 0.5 | 0.5 | | 0.5 | 0.5 | | | | |
| A.H | | 0.5 | | 0.5 | 0.5 | | 0.5 | 0.5 | | | | |
| B.L | | | | | | | | | | | | |
| B.M | | | | 0.5 | 0.5 | | 0.5 | 0.5 | | | | |
| B.H | | | | | | | | | | | | |
| C.L | | | | | | | 0.5 | 0.5 | | | | |
| C.M | | | | | | | 0.5 | 0.5 | | | | |
| C.H | | | | | | | | | | | | |
| D.L | | | | | | | | | | | | |
| D.M | | | | | | | | | | | | |
| D.H | | | | | | | | | | | | |

Fig. 4.4: Adjacency Matrix after a 1st-row scan

At first, no fuzzy-tid-list is formed, generating a fuzzy-tid-list with TID=1 and the minimal fuzzy value for each pair. Figure 4.5 shows the generated Fuzzy-Tid-list after scanning the first row. For the subsequent transaction, the same steps are taken.



| AM-BM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AM-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AM-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-BM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| BM-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| BM-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| BM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| BM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| CL-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

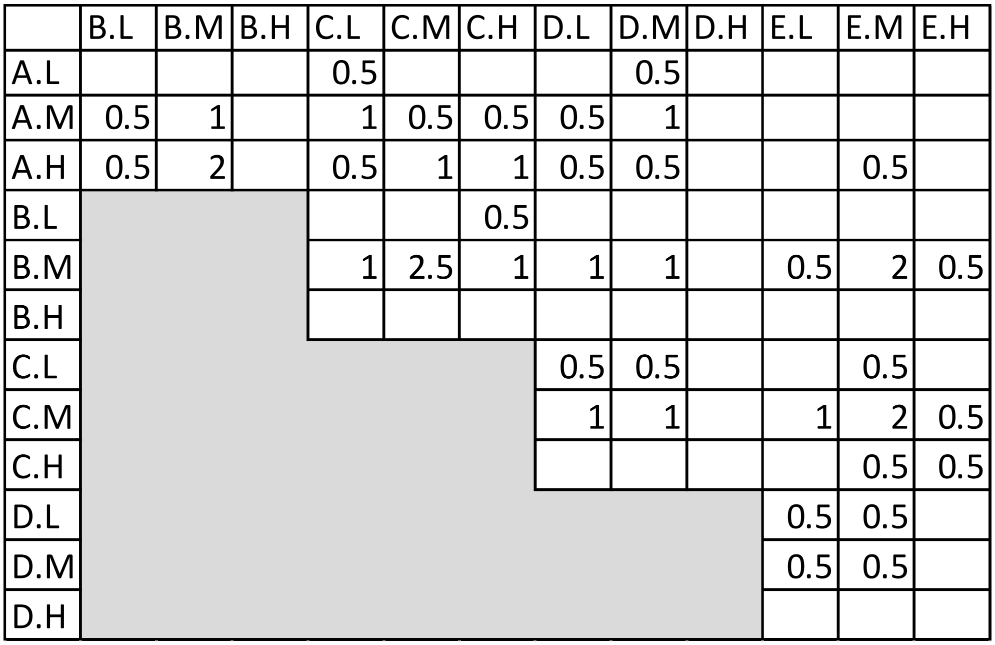
| CL-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| CM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| CM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

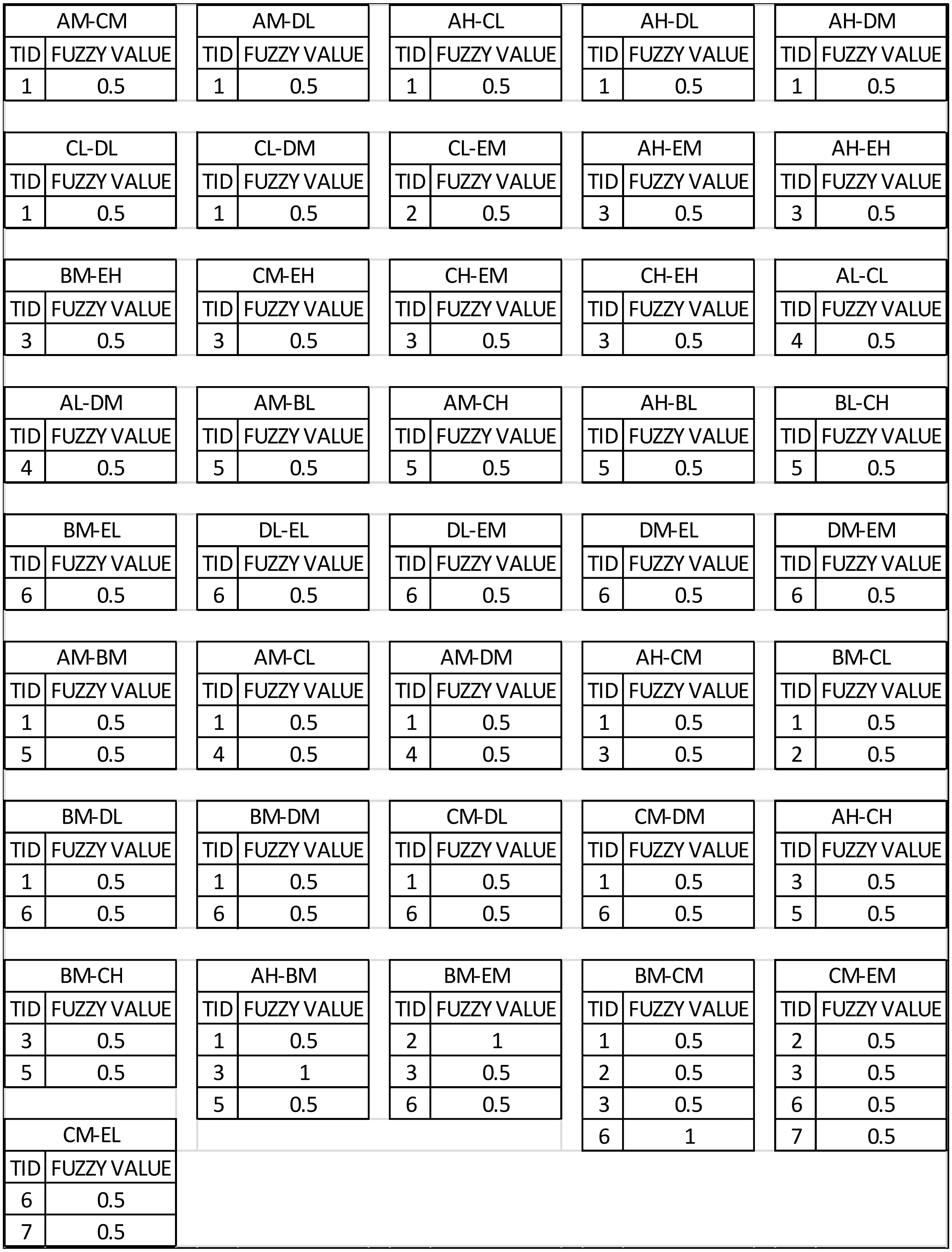
Fig. 4.5: Fuzzy-Tid-list after a 1st-row scan

Following the reading of every transaction, the resulting Adjacency matrix (M) and Fuzzy-Tid-list are illustrated in Figures 4.6 and 4.7, respectively.



| | B.L | B.M | B.H | C.L | C.M | C.H | D.L | D.M | D.H | E.L | E.M | E.H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A.L | | | | 0.5 | | | | 0.5 | | | | |
| A.M | 0.5 | 1 | | 1 | 0.5 | 0.5 | 0.5 | 1 | | | | |
| A.H | 0.5 | 2 | | 0.5 | 1 | 1 | 0.5 | 0.5 | | | 0.5 | |
| B.L | | | | | | 0.5 | | | | | | |
| B.M | | | | 1 | 2.5 | 1 | 1 | 1 | | 0.5 | 2 | 0.5 |
| B.H | | | | | | | | | | | | |
| C.L | | | | | | | 0.5 | 0.5 | | | 0.5 | |
| C.M | | | | | | | 1 | 1 | | 1 | 2 | 0.5 |
| C.H | | | | | | | | | | | 0.5 | 0.5 |
| D.L | | | | | | | | | | 0.5 | 0.5 | |
| D.M | | | | | | | | | | 0.5 | 0.5 | |
| D.H | | | | | | | | | | | | |

Fig. 4.6: Adjacency Matrix after all row scans



| AM-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| AH-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| CL-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| CL-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |

| CL-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.5 |

| AH-EM | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |

| AH-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |

| BM-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |

| CM-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |

| CH-EM | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |

| CH-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |

| AL-CL | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.5 |

| AL-DM | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.5 |

| AM-BL | |
|---|---|
| TID | FUZZY VALUE |
| 5 | 0.5 |

| AM-CH | |
|---|---|
| TID | FUZZY VALUE |
| 5 | 0.5 |

| AH-BL | |
|---|---|
| TID | FUZZY VALUE |
| 5 | 0.5 |

| BL-CH | |
|---|---|
| TID | FUZZY VALUE |
| 5 | 0.5 |

| BM-EL | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.5 |

| DL-EL | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.5 |

| DL-EM | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.5 |

| DM-EL | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.5 |

| DM-EM | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.5 |

| AM-BM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 5 | 0.5 |

| AM-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 4 | 0.5 |

| AM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 4 | 0.5 |

| AH-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 3 | 0.5 |

| BM-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 2 | 0.5 |

| BM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 6 | 0.5 |

| BM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 6 | 0.5 |

| CM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 6 | 0.5 |

| CM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 6 | 0.5 |

| AH-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |
| 5 | 0.5 |

| BM-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.5 |
| 5 | 0.5 |

| AH-BM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 3 | 1 |
| 5 | 0.5 |

| BM-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 1 |
| 3 | 0.5 |
| 6 | 0.5 |

| BM-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.5 |
| 2 | 0.5 |
| 3 | 0.5 |
| 6 | 1 |

| CM-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.5 |
| 3 | 0.5 |
| 6 | 0.5 |
| 7 | 0.5 |

| CM-EL | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.5 |
| 7 | 0.5 |

Fig. 4.7: Fuzzy-Tid-list after all row scan

Here, the generated fuzzy-tid-list is for 2-fuzzy itemsets from each associated pair. Find 2-FFI directly from the constructed adjacency matrix AM.

## 4.2 AMFFI-miner to mine MFFIs Phase 2

In the AMFFI approach, the Adjacency matrix, say M, an upper triangular matrix, cuts down on the enormous number of candidate generations used while creating frequent itemsets. In this phase, fuzzy-tid-lists created in phase 1 extract MFFIs row by row from the adjacency matrix (M). The cell whose value is greater than or equal to the minimum support criterion (δ) should be located by scanning the row starting from M. Fuzzy lists containing 'Row Number-Column Number' of a known cell are retrieved from fuzzy-tid-lists and designated as fuzzy 2-frequent itemsets, such as $FL_2$ of this row. Create fuzzy k-frequent itemsets, such as $FL_k$ (K>2), in a subsequent step by intersection-operating TIDs on $FL_{k-1}$. To quickly locate merged fuzzy lists, use the binary search technique. The construction of a fuzzy list structure for k-frequent itemsets (k>2) is shown in Algorithm 4.2.

***Algorithm 4.2: Fuzzy-Tid-list Construction (from existing fuzzy-Tid-list) for greater than 2-fuzzy itemsets***

```
Input: X.FTL, the fuzzy-tid-list of x; Y.FTL, the fuzzy-tid-list of y
Output: XY. FTL, the fuzzy-tid-list of x and y
Step 1: If x and y belong to the same item, then
        Return null.
Step 2: Initialize fuzzy-tid- list
        XY. FTL ← null;
Step 3: Find common tid from X.FTL and Y.FTL by comparing each element using binary
        search
        For each Ex∈X.FTL do
        If ∃Ey∈Y.FTL and Ex.tid == Ey.tid then
            Exy.tid ← Ex.tid;
            Exy.fv ← min(Ex.fv, Ey.fv );
            Exy←<Exy.tid, Exy.fv>
            Add Exy to XY.FTL.
Step 4: return XY.FTL.
```

Avoid joining fuzzy itemsets that can't create their superset knowing from the adjacency matrix M to limit the search space and candidate set. Take the second row from M for a running example when the minimum support threshold δ = 1. The cells BM, CL, and DM of this row number AM met the minimum support requirement. Therefore, FL2 in this row is AM-BM, AM-CL, and AM-DM. This potential superset is formed by joining AM-BM-CL, AM-BM-DM, and AM-CL-DM from $FL_2$. The proposed technique, however, does not join AM-CL-DM because it is aware that the created superset does not meet the required minimum support level. Fuzzy frequent itemsets AM-CL and AM-DM are present here. The CL row and DM column cell value in the adjacency matrix (M) shown in Figure 4.6 check whether its superset is possible. It might be achievable if it is more significant than or equal to the minimum support value; otherwise, it is impossible. In our example, 0.5, this value does not meet the minimum support threshold, making its superset impossible. Extensions are discarded before joining since they are not fuzzy frequent itemsets. This way, the join operation minimizes the candidate set, improving the running time.

In the same way for joining AM-BM-CL and AM-BM-DM, it is aware that the created superset may meet the required minimum support level. For itemset AM-BM-CL, fuzzy frequent itemsets AM-BM and AM-CL are present here. The BM row and CL column cell value in our example 1.0 meet the minimum support threshold, making its superset possible. For the next itemset, AM-BM-DM, fuzzy frequent itemsets AM-BM and AM-DM are present here. The BM row and DM column cell value in our example 1.0 meet the min support threshold, making its superset possible. After joining AM-BM-CL and AM-BM-DM, know that generated fuzzy itemsets are not frequent. Actually, without an adjacency matrix need to generate three candidate itemsets, but our approach minimizes two candidate itemsets. For this step, AMFFI and AMFFI-miner methods are shown in Algorithms 4.3 and 4.4, respectively.

***Algorithm 4.3: AMFFI method***

```
Input: AM adjacency Matrix; FTLs, fuzzy-tid-list of 2-itemsets; min_supp
Output: MFFIs, the set of multiple fuzzy frequent itemsets.
Step 1: for each row in AM, do
        Initialize fuzzy-tid-list
        L.FTL ← null;
Step 2:  for each cell in the row
          If cell value >= min_supp
                Get fuzzy-tid-list of (F[row.id] [cell.id]) from FTLs into temp
               Add temp to L.FTL
        Call AMFFI-miner with L.FTL
```

***Algorithm 4.4: AMFFI-miner***

```
Input: AM adjacency Matrix; FTLs, fuzzy-tid-list; min_supp
Output: MFFIs, the set of multiple fuzzy frequent itemsets.
Step 1: for each fuzzy-tid-list X in FTLs, do
Step 2: if SUM.X.fv >= min_supp then (where fv = fuzzy value)
                MFFIs ← X ∪ MFFIs.
Step 3:temp.FTLs ← null;
Step 4: for each fuzzy-tid-list Y after X in FTLs, do
        If X.ITEM = Y.ITEM, then
                Continue;
        If AM [X.ITEM] [Y.ITEM] >= min_supp then
             temp.FTL ← temp.FTLs + Construct(X, Y );
Step 5: AMFFI-Miner (temp.FTLs);
Step 6: Return MFFIs.
```

From the above discussed example, the scenario for the join operations in AMFFI technique are shown in Table 4.3. It shows that the total number of possible joins is 18; while the AMFFI technique generates only four. Among them two are multiple fuzzy frequent itemsets, so the AMFFI approach generates only two unnecessary candidate itemsets. Generated 3-MFFIs are AH-BM-CM, AH-BM-CH. Extensions are discarded before joining since they are not fuzzy frequent itemsets directly from the adjacency matrix. This way, the join operation minimizes the candidate set, improving the running time performance.

Table 4.3: Scenario of AMFFI for illustrative example

| **Row** | **Possible itemsets** | **Total Possible** | **Pruned itemsets before joining** | **Total Joins** | **No. of $FL_3$ Itemsets** |
|---|---|---|---|---|---|
| AM | AM-BM-CL, AM-BM-DM, AM-CL-DM | 3 | AM-CL-DM | 2 | 0 |
| AH | AH-BM-CM, AH-BM-CH | 2 | Nil | 2 | 2 |
| BM | BM-CL-DL, BM-CL-DM, BM-CL-EM, BM-CH-DL, BM-CH-DM, BM-CH-EM, BM-DL-EM, BM-DM-EM | 8 | ALL | 0 | - |
| CM | CM-DL-EL, CM-DL-EM, CM-DM-EL, CM-DM-EM | 4 | ALL | 0 | - |

The subsequent chapter discusses performance comparisons of the AMFFI technique with state-of-the-art methods for running time, memory utilization, and node join counts.

# CHAPTER 5

# Performance Evaluation of AMFFI

This chapter describes how the performance of the proposed method called AMFFI concerning the state-of-the-art MFFI-Miner [13] method. The authors make a comparison of their method MFFI-miner concerning the state-of-the-art GDF [47] and the UBMFFP tree [41] methods. The proposed AMFFI and MFFI-miner methods were implemented in Java. Experiments were evaluated to check performance on three real-life datasets, chess [62], mushroom [62], and Chicago_crime_2001-2017 from UCI learning as well as two synthetic datasets, T10I4D100k [62] and work [63] datasets from SPMF—details of datasets given in following Table 5.1. In the datasets, the item quantities were arbitrarily distributed in intervals between 1 and 7.

Table 5.1 Details of datasets

| Sr. No. | Dataset | No. Of. Transactions | No. of Items |
|---|---|---|---|
| 1 | Chess | 3196 | 75 |
| 2 | Mushroom | 8416 | 119 |
| 3 | Chicago_crime_2001-2017 | 2662309 | 35 |
| 4 | T10I4D100k | 100000 | 1000 |
| 5 | Synthetic (work) | 600000 | 600 |

AMFFI and MFFI-miner [13] were implemented for the 3-term membership function and the 5-term membership function. The experiment's runtime, memory utilization, and node join counts are assessed for comparison against the planned approaches. Section 5.1 discusses performance evaluation for the 3-term membership function of the proposed approach AMFFI with state-of-the-art methods on standard real and synthetic datasets. Section 5.2 discusses performance evaluation for the 5-term membership function of the proposed approach AMFFI with state-of-the-art methods on standard real datasets.

## 5.1 Performance evaluation for 3-term membership function

### 5.1.1 Runtime Analysis for 3-term member function:

The implemented 3-term fuzzy linguistic AMFFI and MFFI-miner [13] were evaluated with different min-support thresholds to compare execution time. The output of execution time evaluated on chess, mushroom, Chicago_crime_2001-2017, T10I4D100k, and work (synthetic) datasets are shown in Figure 5.1, Figure 5.2, Figure 5.3, Figure 5.4, and Figure 5.5, respectively.

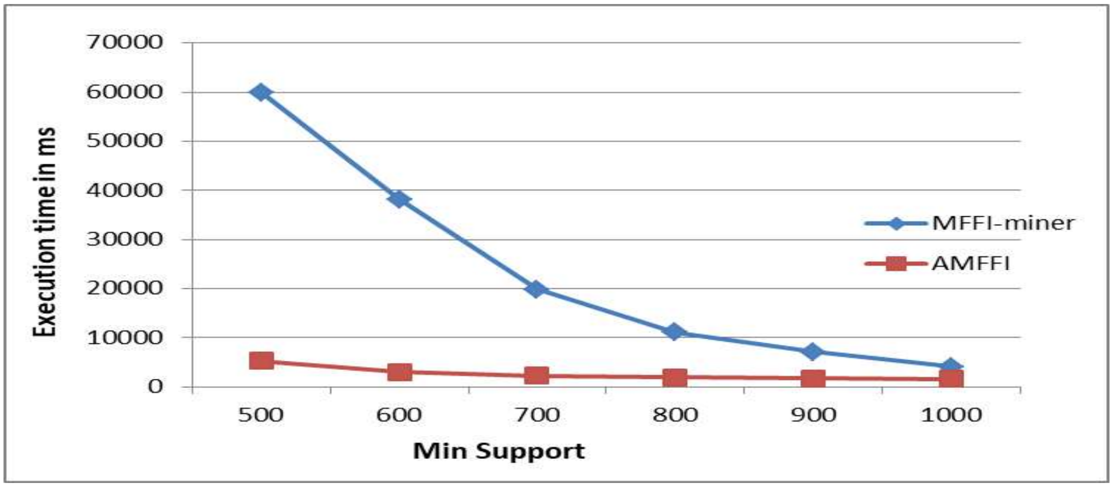


Fig. 5.1: Performance Evaluation for running time on Chess Dataset for 3-term member function

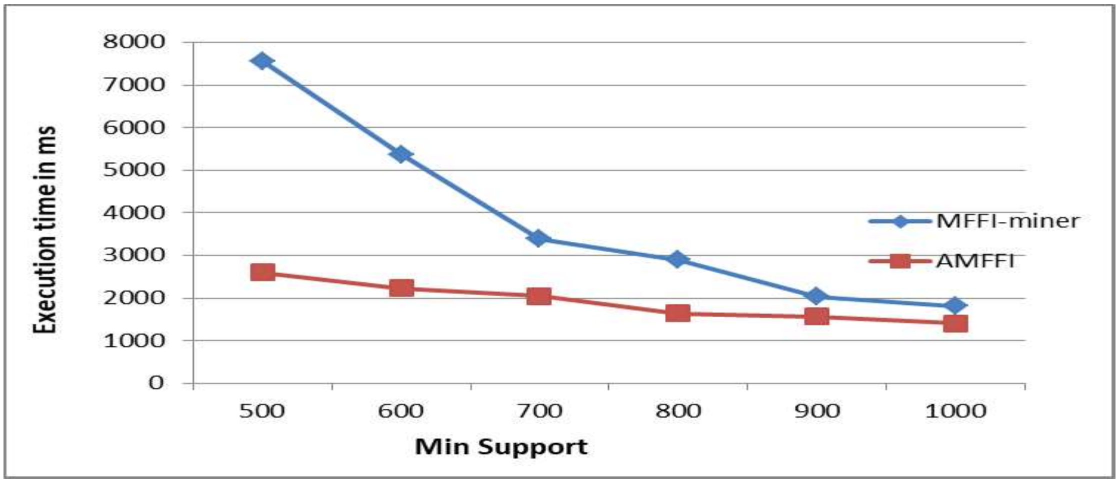


Fig. 5.2: Performance Evaluation for running time on Mushroom Dataset for 3-term member function

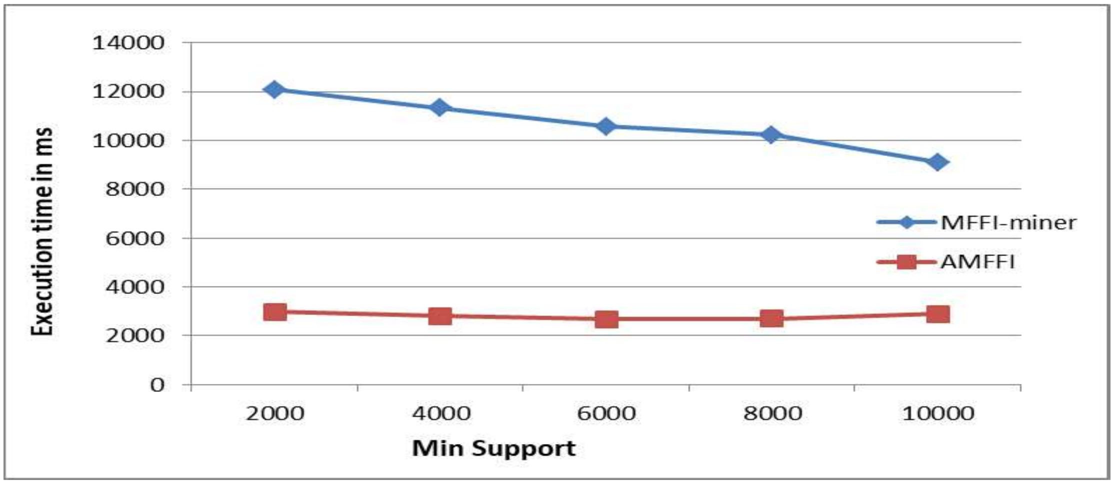


Fig. 5.3: Performance Evaluation for running time on Chicago_crime_2001-2017 Dataset for 3-term member function

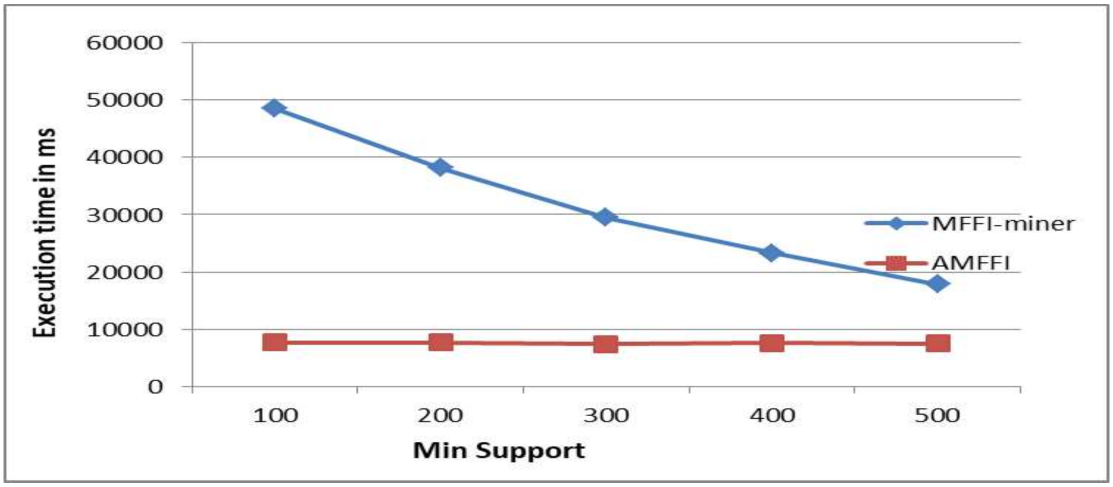


Fig. 5.4: Performance Evaluation for running time on T10I4D100k Dataset for 3-term member function

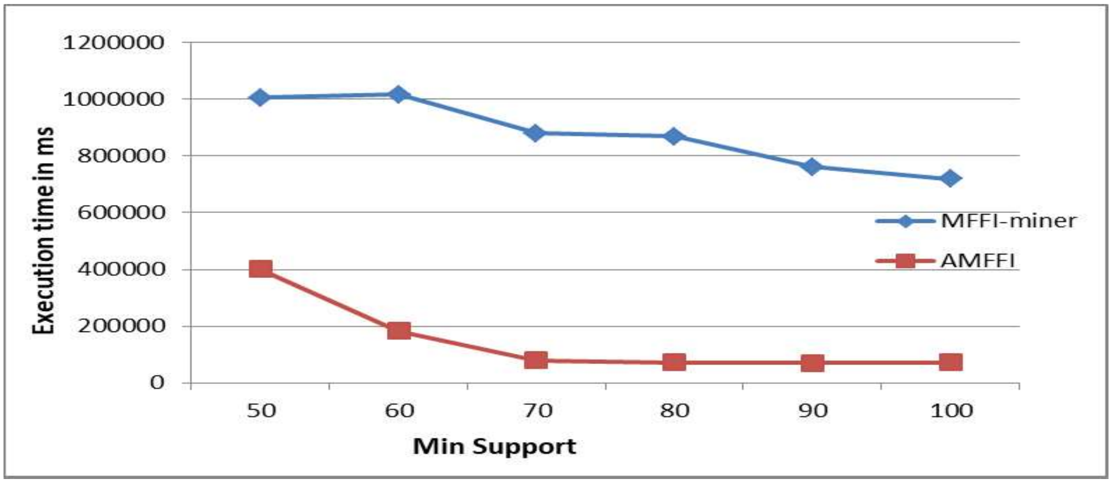


Fig. 5.5: Performance Evaluation for running time on Synthetic (work) Dataset for 3-term member function

MFFI-miner [13] authors show that the running time performance of its proposed method is good concerning GDF [47] and the UBMFFP tree [41]. The result shows that the performance of the proposed AMFFI method is faster than the existing MFFI-miner method. The result also indicates that the AMFFI method outperformed the current method when taking a lower min-support threshold.

### 5.1.2 Join counts Analysis for 3-term member function:

This section evaluates performance for the number of join counts that occur when generating MFFIs. The output of the number of join counts generated while evaluating the chess, mushroom, and T10I4D100k datasets are shown in Figure 5.6, Figure 5.7, Figure 5.8, Figure 5.9, and Figure 5.10, respectively.

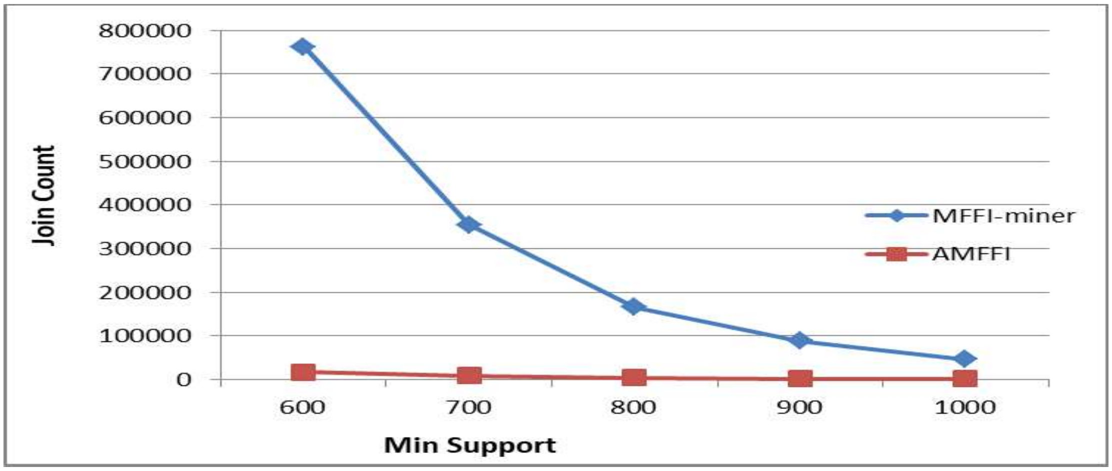


Fig. 5.6: Performance Evaluation for join counts on Chess Dataset for 3-term member function

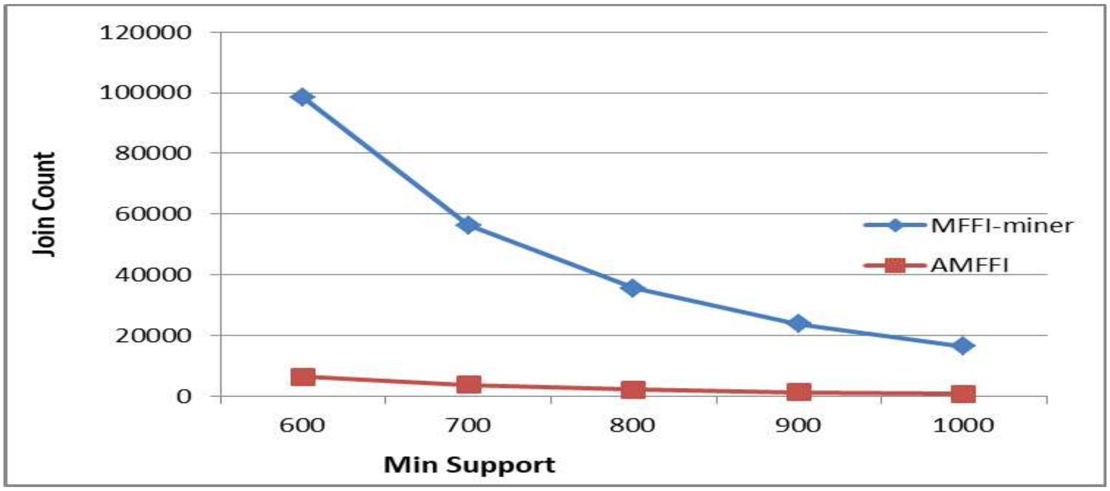


Fig. 5.7: Performance Evaluation for join counts on Mushroom Dataset for 3-term member function

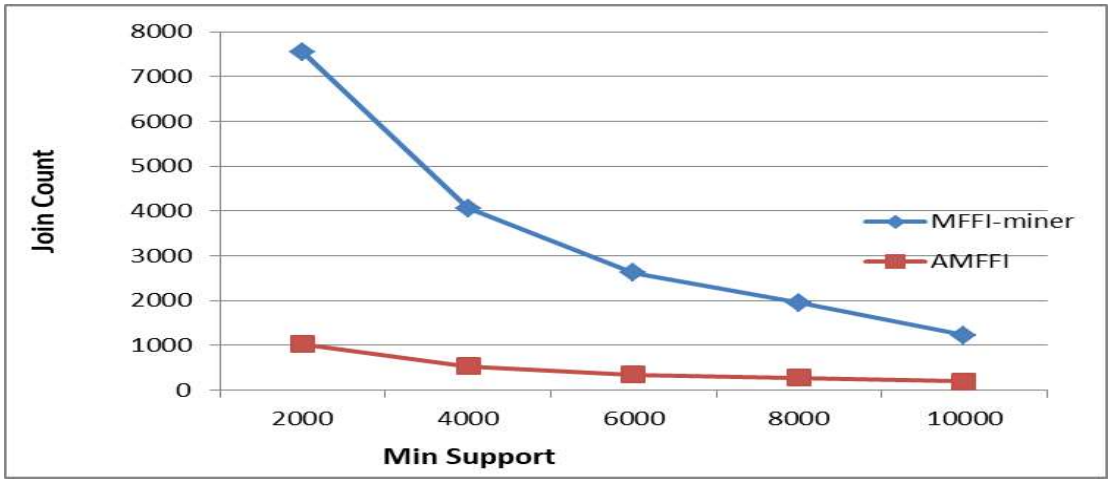


Fig. 5.8: Performance Evaluation for join counts on Chicago_crime_2001-2017 Dataset for 3-term member function

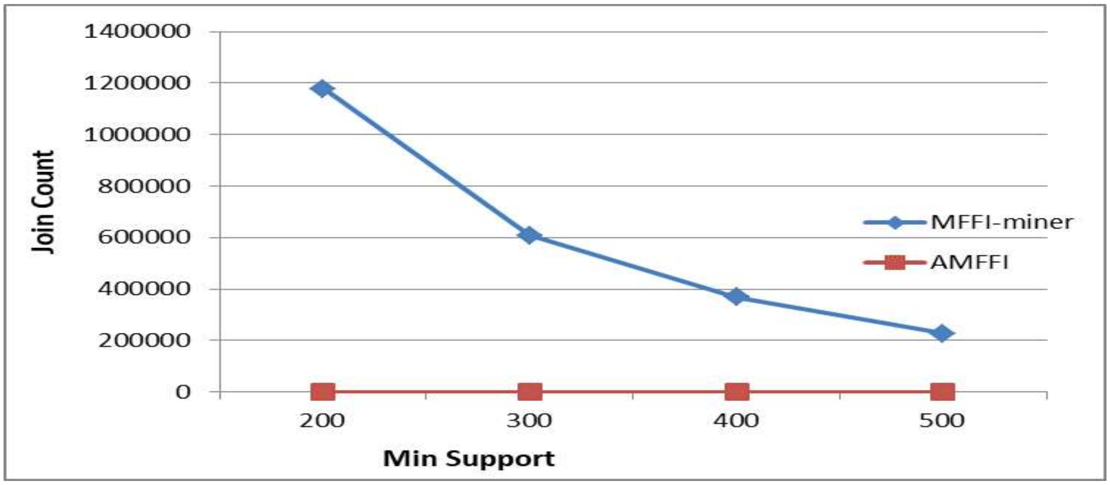


Fig. 5.9: Performance Evaluation for join counts on T10I4D100k Dataset for 3-term member function

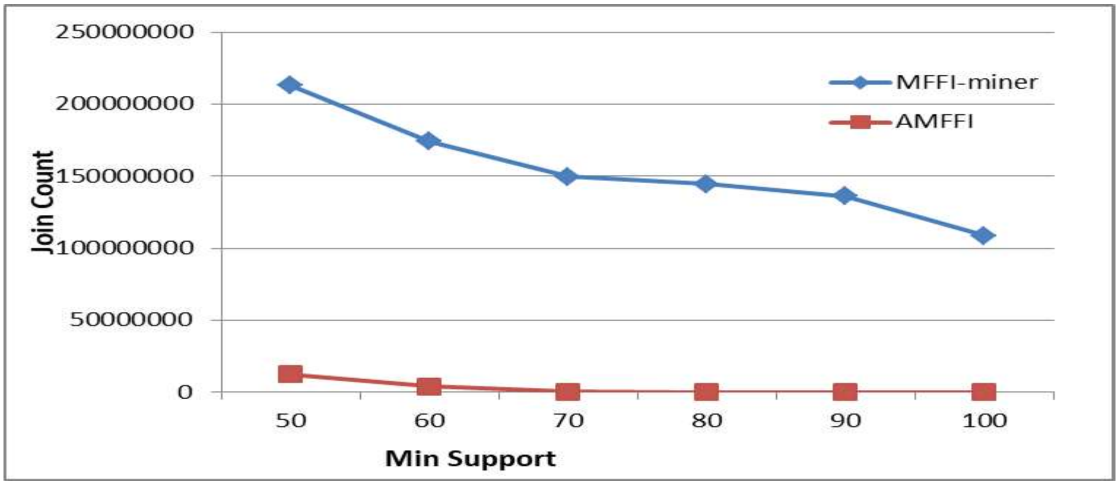


Fig. 5.10: Performance Evaluation for join counts on synthetic (work) Dataset for 3-term member function

The result shows that the AMFFI method generates fewer join counts (candidate itemsets), and provide most impressive performance compared to cutting-edge approaches.

### 5.1.3 Memory Usage Analysis for 3-term member function:

In this section, performance concerning memory utilization is evaluated when evaluating experiments. The output of memory usage while evaluating the investigation on the chess, mushroom, and T10I4D100k datasets are shown in Figure 5.11, Figure 5.12, Figure 5.13, Figure 5.14, and Figure 5.15, respectively.

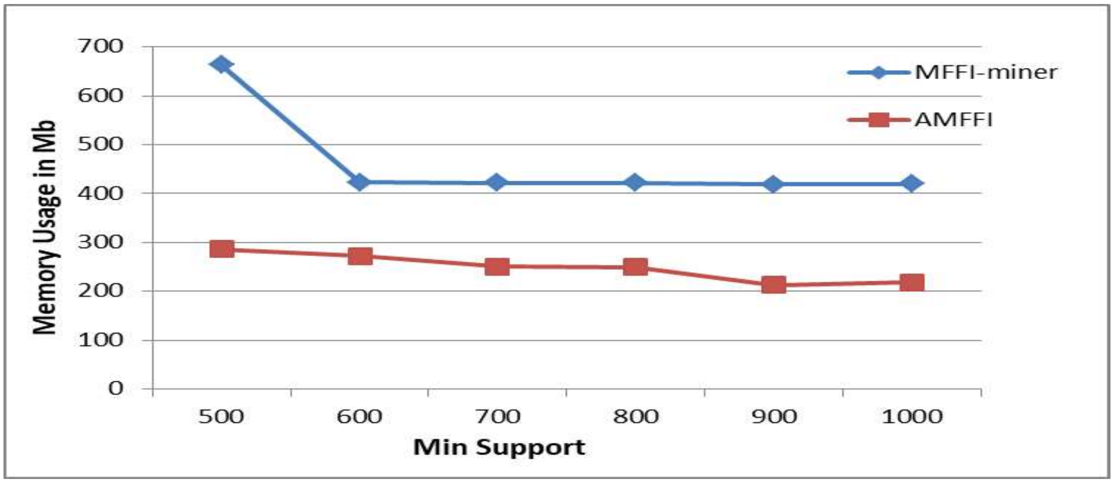


Fig. 5.11: Performance Evaluation for memory usage on Chess Dataset for 3-term member function

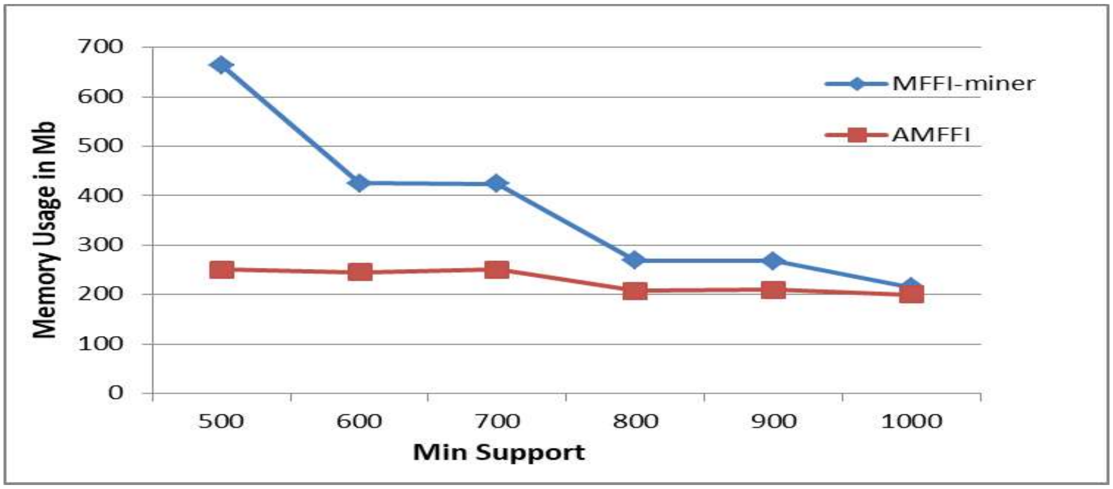


Fig. 5.12: Performance Evaluation for memory usage on Mushroom Dataset for 3-term member function

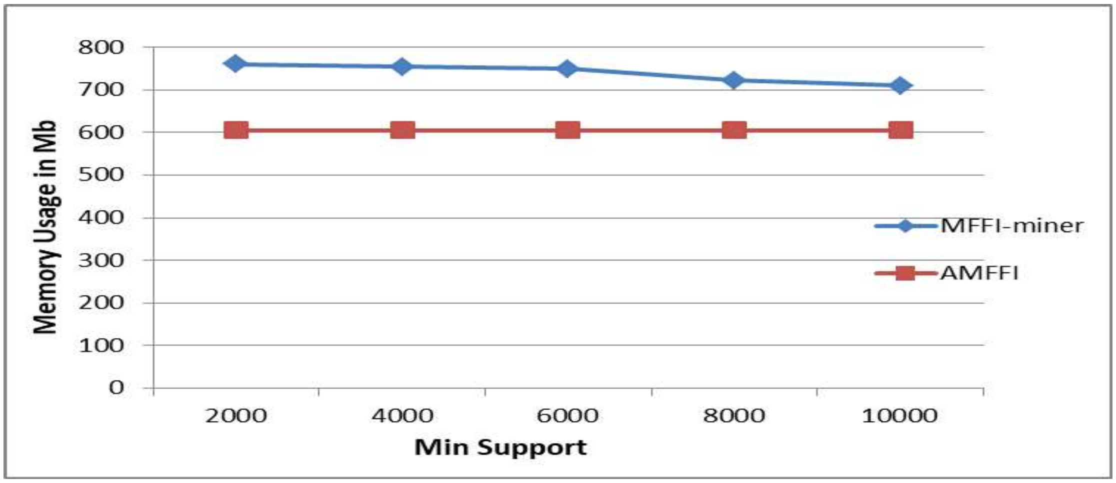


Fig. 5.13: Performance Evaluation for memory usage on Chicago_crime_2001-2017 Dataset for 3-term member function

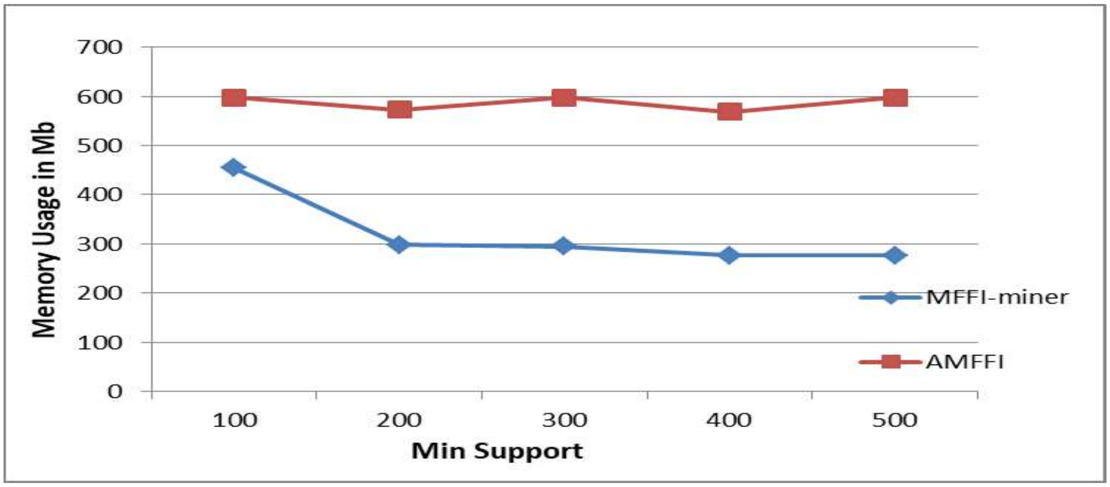


Fig. 5.14: Performance Evaluation for memory usage on T10I4D100k Dataset for 3-term member function

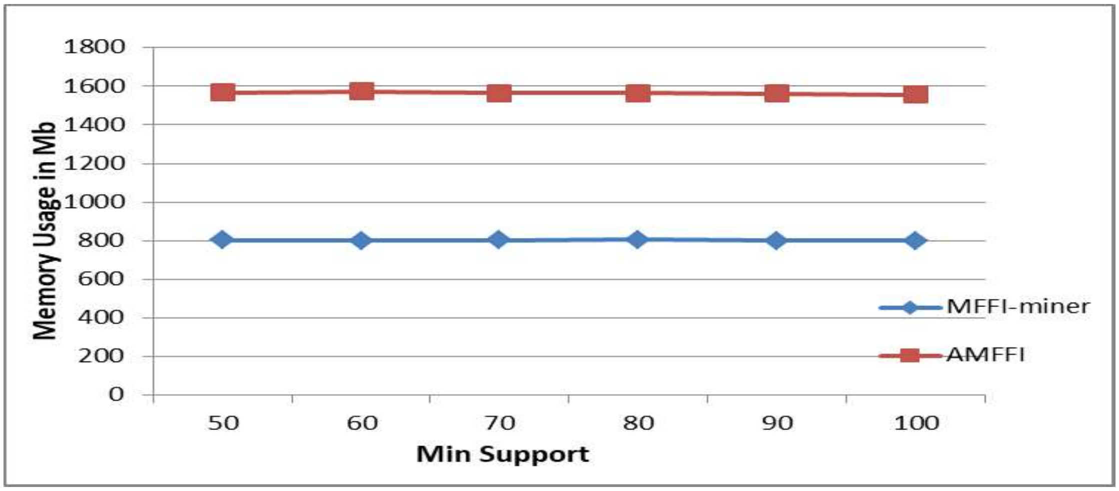


Fig. 5.15: Performance Evaluation for memory usage on Synthetic (work) Dataset for 3-term member function

The outcome demonstrates that compared to the current MFFI-miner approach, the chess mushroom and Chicago_crime_2001-2017 dataset AMFFI method uses less memory. Furthermore, it has been found that the synthetic work and T10I4D100k dataset AMFFI method uses more memory than the MFFI-miner approach. Alternative trials using diverse datasets demonstrate that the suggested approach demands increased memory capacity when the quantity of items surpasses 450.

## 5.2 Performance evaluation for 5-term membership function

### 5.2.1 Runtime Analysis for 5-term member function:

The implemented 5-term fuzzy linguistic AMFFI and MFFI-miner [13] were evaluated with different min-support thresholds to compare execution running time. The output of execution running time evaluated on chess, mushroom, and Chicago_crime_2001-2017 datasets is shown in Figure 5.16, Figure 5.17, and Figure 5.18, respectively.

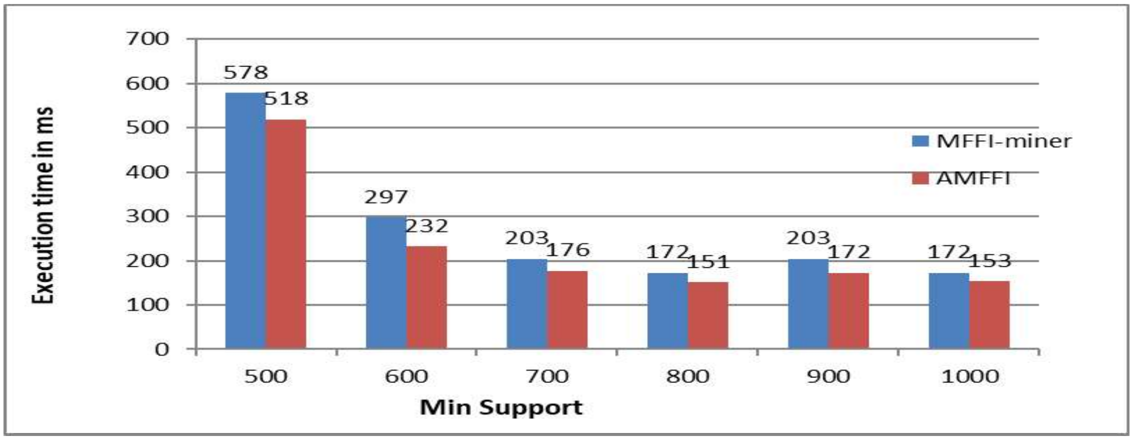


Fig. 5.16: Performance Evaluation for running time on Chess Dataset for 5-term member function

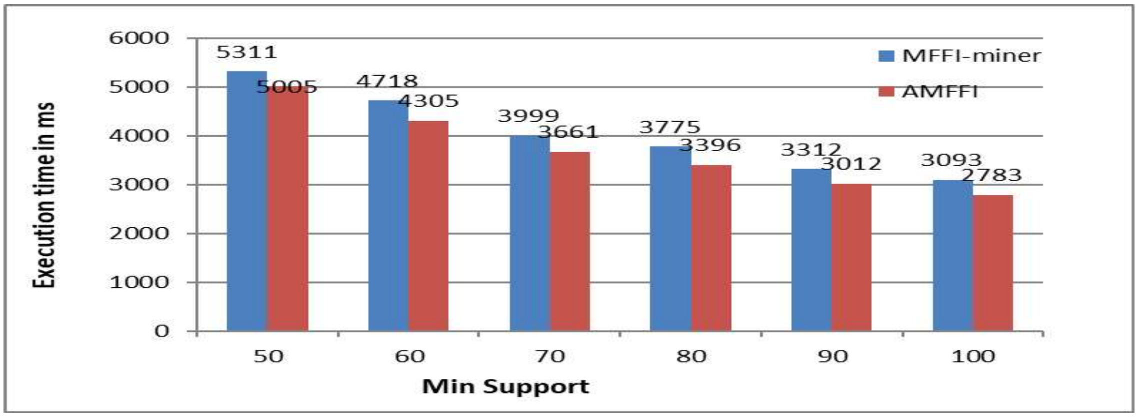


Fig. 5.17: Performance Evaluation for running time on Mushroom Dataset for 5-term member function

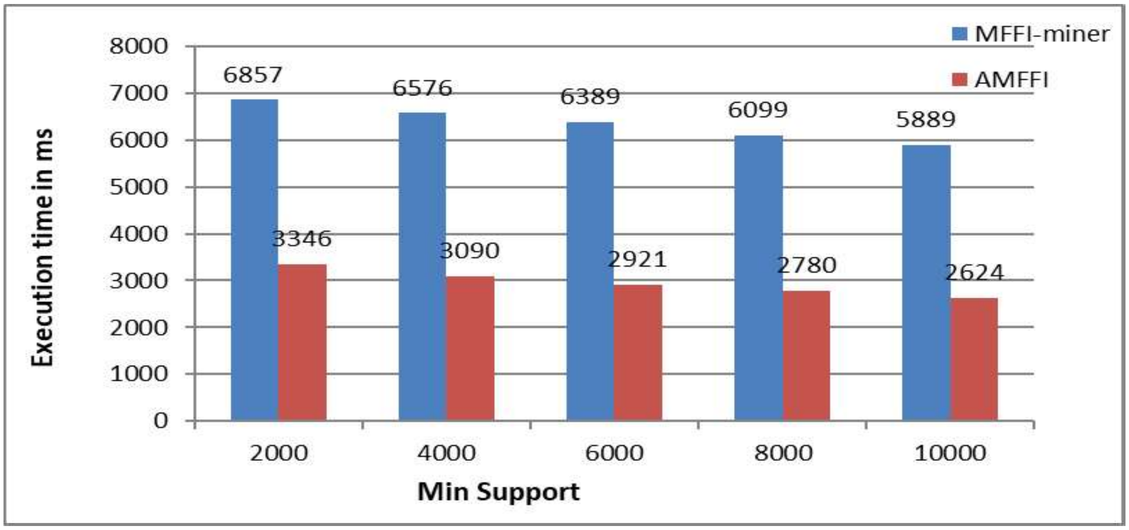


Fig. 5.18: Performance Evaluation for running time on Chicago_crime_2001-2017 Dataset for 5-term member function

The result shows that the performance of the proposed AMFFI method is faster than the existing MFFI-miner method. The result also indicates that the AMFFI method outperformed the current method when taking a lower min-support threshold.

### 5.2.2 Join counts Analysis for 5-term member function:

This section evaluates performance for the number of join counts that occur when generating MFFIs. The output of the number of join counts generated while evaluating the chess, mushroom, and Chicago_crime_2001-2017 datasets are shown in Figure 5.19, Figure 5.20, and Figure 5.21, respectively.

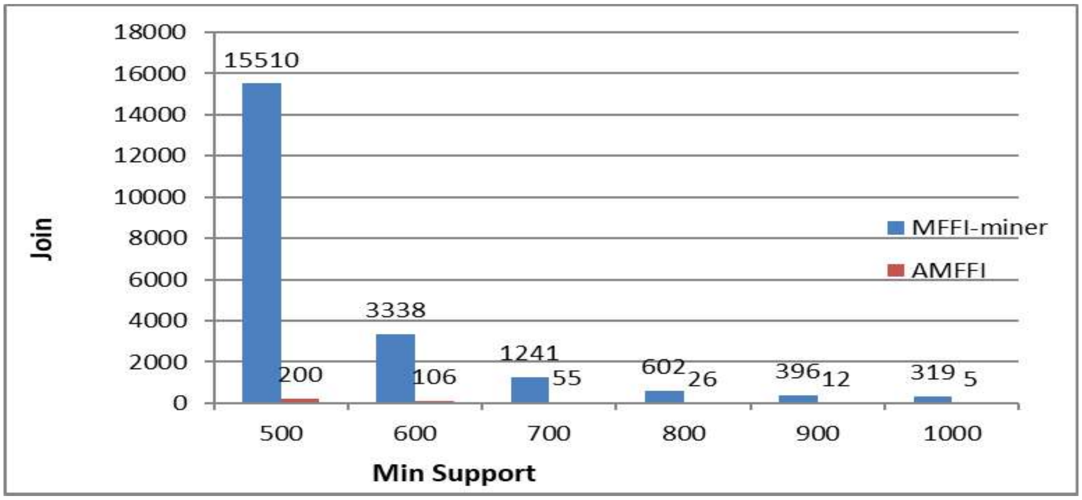


Fig. 5.19: Performance Evaluation for join counts on Chess Dataset for 5-term member function

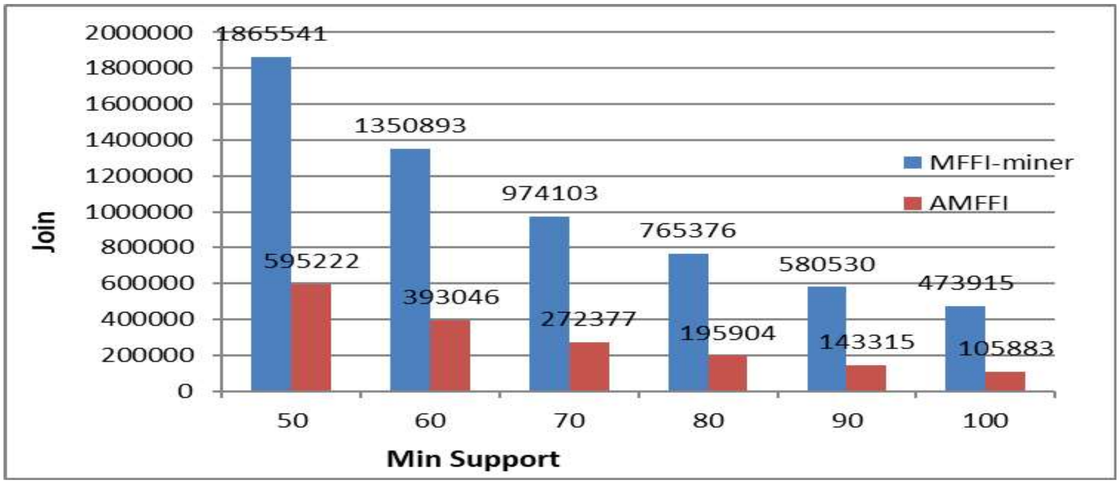


Fig. 5.20: Performance Evaluation for join counts on Mushroom Dataset for 5-term member function

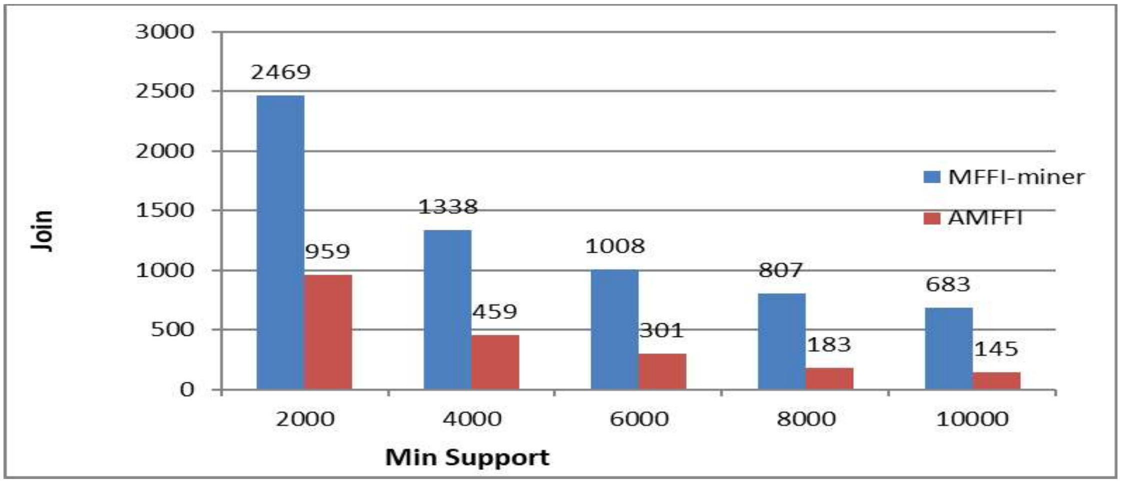


Fig. 5.21: Performance Evaluation for join counts on Chicago_crime_2001-2017 Dataset for 5-term member function

The result shows that the AMFFI method generates fewer join counts (candidate itemsets). Additionally, it has been noted that the AMFFI method's join count performance is the most impressive. Compared to cutting-edge approaches, the suggested AMFFI method produces fewer candidate itemsets.

### 5.2.3 Memory Usage Analysis for 5-term member function:

In this section, performance concerning memory utilization is evaluated when evaluating experiments. The output of memory usage while evaluating the investigation on the chess, mushroom, and Chicago_crime_2001-2017 datasets are shown in Figure 5.22, 5.23, and 5.24, respectively.

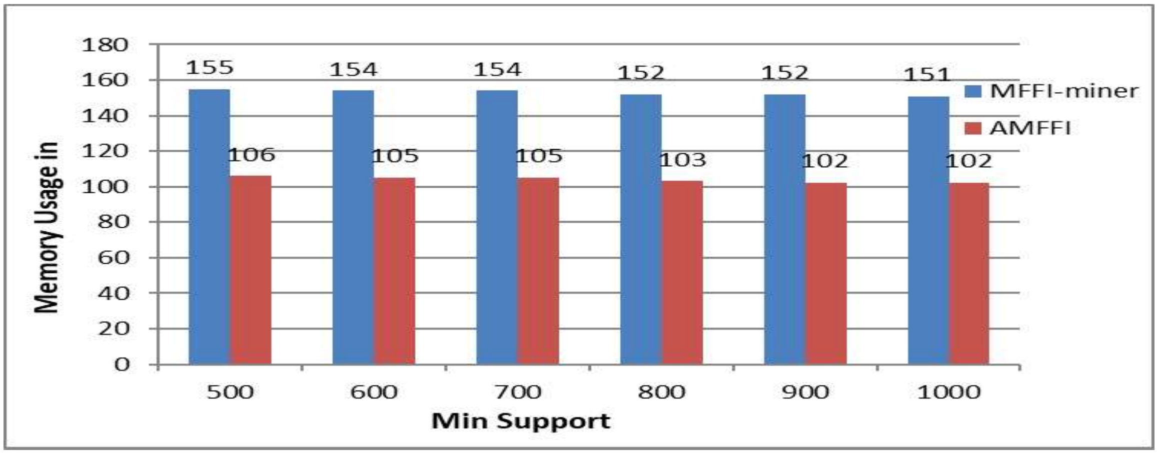


Fig. 5.22: Performance Evaluation for memory usage on Chess Dataset for 5-term member function

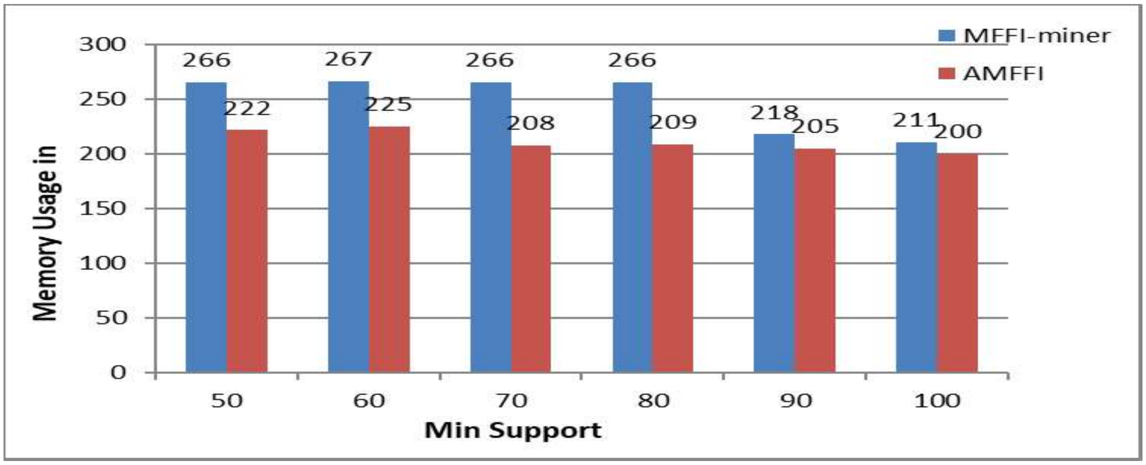


Fig. 5.23: Performance Evaluation for memory usage on Mushroom Dataset for 5-term member function

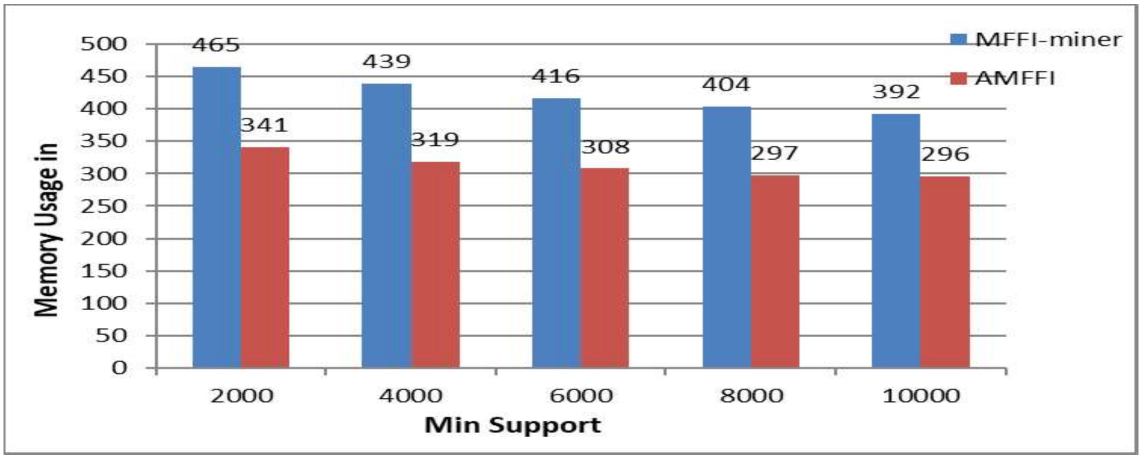


Fig. 5.24: Performance Evaluation for memory usage on Chicago_crime_2001-2017 Dataset for 5-term member function

The outcome demonstrates that compared to the current MFFI-miner approach, the chess mushroom and Chicago_crime_2001-2017 dataset AMFFI method uses less memory.

## 5.3 Overall improvement in percentage

### 5.3.1 Overall improvement in the percentage of 3-term member function

The implemented 3-term fuzzy linguistic AMFFI and MFFI-miner [13] were evaluated with different min-support thresholds to compare execution running time, number of join counts, and memory utilization. The overall improvement of AMFFI vs. MFFI-miner in percentage with different min-support on chess, mushroom, Chicago_crime_2001-2017, T10I4D100k, and work (synthetic) dataset are shown in Table 5.2, Table 5.3, Table 5.4, Table 5.5, and Table 5.6, respectively.

Table 5.2: Improvement on chess dataset for 3-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 500 | 97.79 | 91.21 | 56.86 |
| 600 | 97.80 | 91.98 | 35.70 |
| 700 | 97.64 | 88.51 | 40.52 |
| 800 | 97.37 | 82.00 | 41.00 |
| 900 | 97.34 | 75.03 | 49.16 |
| 1000 | 97.09 | 60.07 | 48.10 |

Table 5.3: Improvement on Mushroom dataset for 3-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 500 | 93.15 | 65.68 | 62.26 |
| 600 | 93.50 | 58.44 | 42.35 |
| 700 | 93.39 | 39.69 | 40.80 |
| 800 | 93.96 | 43.12 | 23.05 |
| 900 | 94.61 | 23.02 | 21.64 |
| 1000 | 94.91 | 22.49 | 06.98 |

Table 5.4: Improvement on Chicago_crime_2001-2017 dataset for 3-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 2000 | 86.40 | 75.30 | 20.50 |
| 4000 | 86.78 | 75.17 | 19.76 |
| 6000 | 86.56 | 74.74 | 19.23 |
| 8000 | 85.67 | 73.59 | 16.32 |
| 10000 | 84.00 | 68.27 | 14.79 |

Table 5.5: Improvement on T10I4D100k dataset for 3-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 100 | 99.9 | 84.21 | -31.72 |
| 200 | 99.98 | 79.86 | -92.28 |
| 300 | 99.99 | 75.04 | -102.71 |
| 400 | 100 | 67.6 | -106.16 |
| 500 | 100 | 58.16 | -116.67 |

Table 5.6: Improvement of Synthetic (Work) dataset for 3-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 50 | 94.28 | 59.99 | -95.76 |
| 60 | 97.70 | 82.07 | -96.50 |
| 70 | 99.95 | 90.97 | -95.26 |
| 80 | 100.0 | 91.76 | -94.53 |
| 90 | 100.0 | 90.67 | -95.25 |
| 100 | 100.0 | 89.89 | -94.38 |

### 5.3.2 Overall improvement in percentage on 5-term member function

The implemented 5-term fuzzy linguistic AMFFI and MFFI-miner [13] were evaluated with different min-support thresholds to compare execution running time, number of join counts, and memory utilization. The overall improvement of AMFFI vs. MFFI-miner in percentage with different min-support on chess, mushroom, and Chicago_crime_2001-2017 datasets are shown in Table 5.7, Table 5.8, and Table 5.9, respectively.

Table 5.7: Improvement on chess dataset for 5-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 500 | 98.71 | 10.38 | 31.61 |
| 600 | 96.82 | 21.89 | 31.82 |
| 700 | 95.57 | 13.30 | 31.82 |
| 800 | 95.68 | 12.21 | 32.24 |
| 900 | 96.97 | 15.27 | 32.89 |
| 1000 | 98.43 | 11.05 | 32.45 |

Table 5.8: Improvement on Mushroom dataset for 5-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 500 | 68.09 | 05.76 | 16.54 |
| 600 | 70.90 | 08.75 | 15.73 |
| 700 | 72.04 | 08.45 | 21.80 |
| 800 | 74.40 | 10.04 | 21.43 |
| 900 | 75.31 | 09.06 | 05.96 |
| 1000 | 77.66 | 10.02 | 05.21 |

Table 5.9: Improvement on Chicago_crime_2001-2017 dataset for 5-term member function

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 2000 | 61.16 | 51.20 | 26.67 |
| 4000 | 65.70 | 53.01 | 27.33 |
| 6000 | 70.14 | 54.28 | 25.96 |
| 8000 | 77.32 | 54.42 | 26.49 |
| 10000 | 78.77 | 55.44 | 24.49 |

The above tables 5.2 to 5.9 demonstrate that AMFFI approach is better than current MFFI-miner approach with respect to execution time and node join counts.

# CHAPTER 6

# An Efficient Multiple Fuzzy Frequent Patterns Mining with Adjacency matrix and Type-2 Member function

The main contribution of this research is to design an efficient multiple fuzzy frequent itemsets mining method using an adjacency matrix and fuzzy-tid-list. Proposed methods Multiple Fuzzy Frequent Patterns Mining with Adjacency matrix and Type-2 Member function (MFFPA-2) scan database only once and reduce the number of nodes join counts (candidate itemsets) by pruning non-frequent itemsets extracted from the adjacency matrix.

The research methodology comprises developing a novel approach to mining MFFIs using an adjacency matrix and fuzzy-tid-list structures. The MFFPA-2 proposed model producing MFFIs in two step as shown in Figure 6.1.

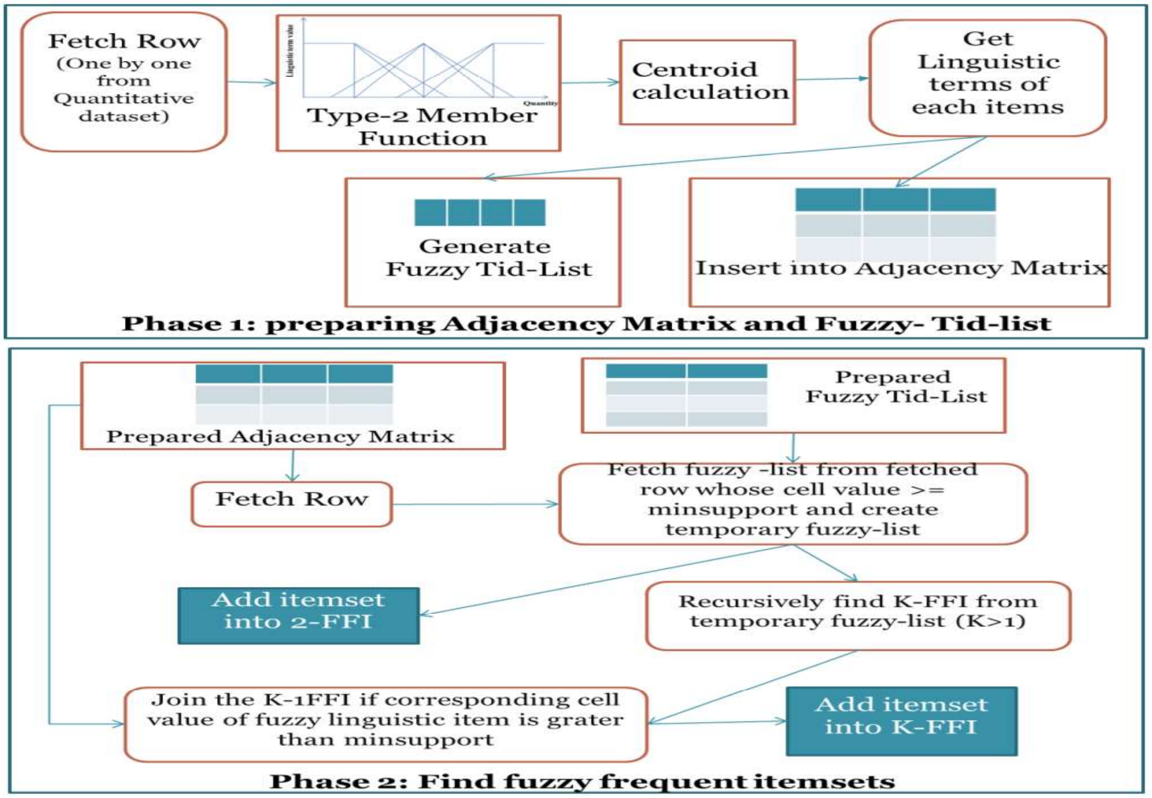

Fetch Row
(One by one from Quantitative dataset)
Linguistic term value
Quantity
Type-2 Member Function
Centroid calculation
Get Linguistic terms of each items
Generate Fuzzy Tid-List
Insert into Adjacency Matrix
Phase 1: preparing Adjacency Matrix and Fuzzy- Tid-list
Prepared Adjacency Matrix
Prepared Fuzzy Tid-List
Fetch Row
Fetch fuzzy -list from fetched row whose cell value >= minsupport and create temporary fuzzy-list
Add itemset into 2-FFI
Recursively find K-FFI from temporary fuzzy-list (K>1)
Join the K-1FFI if corresponding cell value of fuzzy linguistic item is grater than minsupport
Add itemset into K-FFI
Phase 2: Find fuzzy frequent itemsets

Fig 6.1: MFFPA-2 proposed model

This chapter suggests a two-step process for creating many fuzzy frequent itemsets. In this technique, use the Type-2 membership function. Type-2 membership functions allow for a more excellent representation of uncertainty. The additional modeling capability of type-2 membership functions can improve decision-making processes when there is a higher level of uncertainty. By considering multiple possibilities and capturing more complex relationships between variables, type-2 fuzzy sets can provide more informed and flexible decision-making.

Using the quantitative dataset D, generate an adjacency matrix and Fuzzy-tid-list in phase 1. Using the MFFPA-2 approach, quickly extract multiple fuzzy frequent itemsets from the adjacency matrix and fuzzy-tid-list discussed in phase 2. The suggested technique effectively creates entire MFFIs from a single database scan. Chapter 7 discusses the performance evaluation of the proposed approach MFFPA-2 with state-of-the-art methods on standard real life datasets.

## 6.1 Adjacency matrix and Fuzzy-tid-list construction

During the first phase, the algorithm transforms the quantitative transaction values into a fuzzy set using the member function with several linguistic terms. In this approach, use the type-2 membership function.

Consider the membership function '£2' for Low, Middle, and High linguistic terms. Adjacency matrix AdjMat (M) of size (m*3) X (m*3) should first be constructed, where 'm' represents the overall count of items in the original dataset D. Total items in D' (fuzzy dataset) is a product of 'm' in D, and 't' linguistic terms of type-2 membership function.

Next, in this phase, scan transaction $T_q$ from quantitative database D shown in Table 6.1 and apply a pre-defined type-2 membership function $£_2$ depicted in Figure 6.2, producing fuzzy linguistic terms as illustrated in Table 6.2.

Table 6.1: Quantitative sample Dataset

| TID | Item with Quantity |
|---|---|
| 1 | A-4, B-3, C-2, D-2 |
| 2 | B-3, C-2, E-3 |
| 3 | A-5, B-3, C-4, E-4 |
| 4 | A-2, C-1, D-3 |
| 5 | A-4, B-2, C-5 |
| 6 | B-3, C-3, D-2, E-2 |
| 7 | C-3, E-2 |

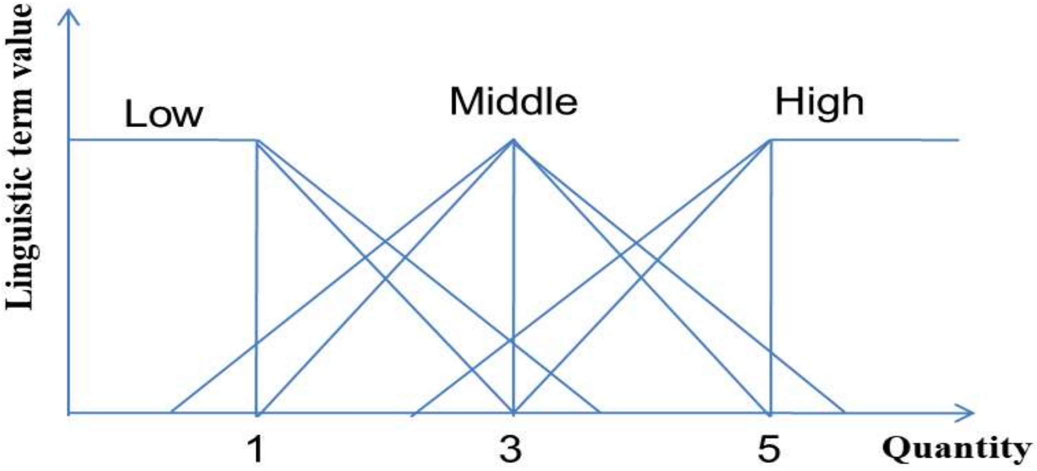
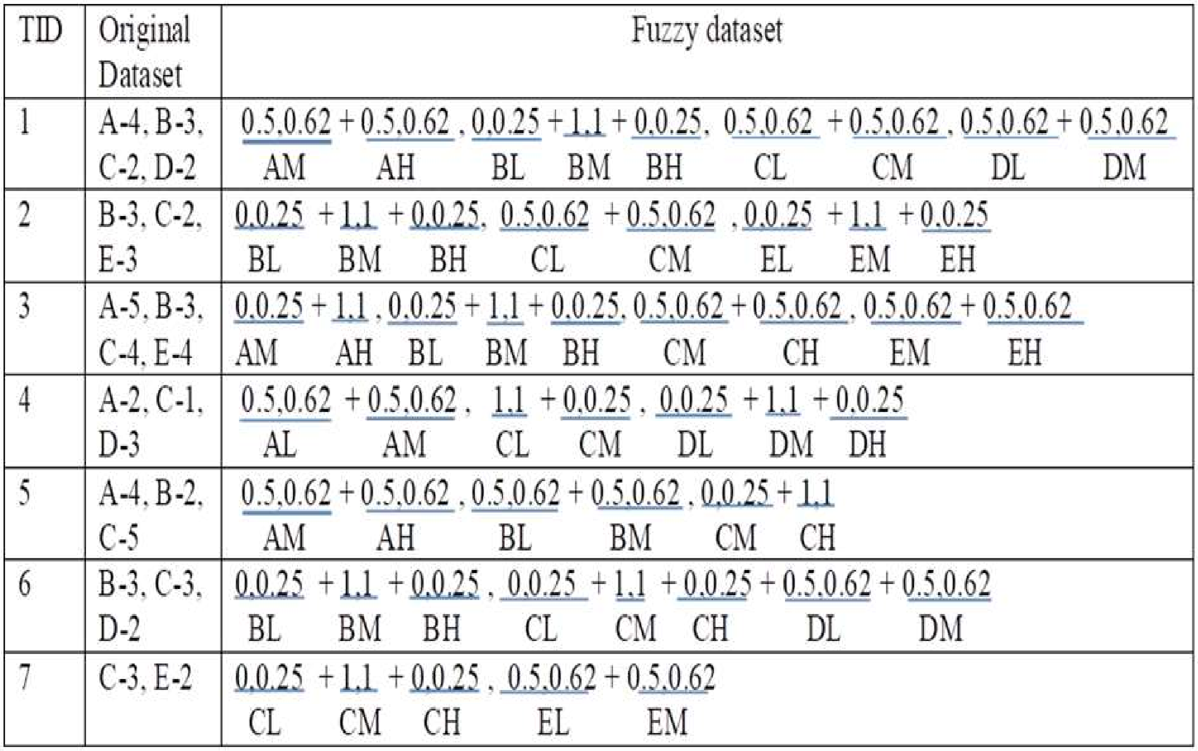


Fig. 6.2: Type-2 Membership Function

Here are two values of each fuzzy linguistic term: the first is associated with a lower boundary, and the second is associated with a higher boundary of membership function, as shown in Figure 6.2.

Table 6.2: Transpose fuzzy dataset using type-2 function



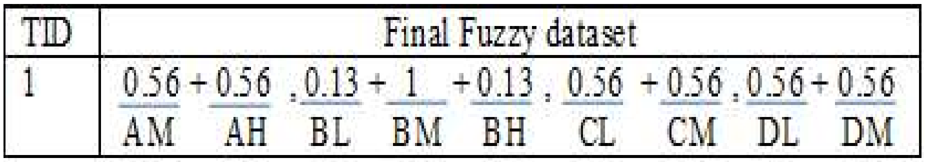
| TID | Original Dataset | Fuzzy dataset |
|---|---|---|
| 1 | A-4, B-3, C-2, D-2 | $\frac{0.5,0.62}{AM}+\frac{0.5,0.62}{AH}, \frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CL}+\frac{0.5,0.62}{CM}, \frac{0.5,0.62}{DL}+\frac{0.5,0.62}{DM}$ |
| 2 | B-3, C-2, E-3 | $\frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CL}+\frac{0.5,0.62}{CM}, \frac{0,0.25}{EL}+\frac{1,1}{EM}+\frac{0,0.25}{EH}$ |
| 3 | A-5, B-3, C-4, E-4 | $\frac{0,0.25}{AM}+\frac{1,1}{AH}, \frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CM}+\frac{0.5,0.62}{CH}, \frac{0.5,0.62}{EM}+\frac{0.5,0.62}{EH}$ |
| 4 | A-2, C-1, D-3 | $\frac{0.5,0.62}{AL}+\frac{0.5,0.62}{AM}, \frac{1,1}{CL}+\frac{0,0.25}{CM}, \frac{0,0.25}{DL}+\frac{1,1}{DM}+\frac{0,0.25}{DH}$ |
| 5 | A-4, B-2, C-5 | $\frac{0.5,0.62}{AM}+\frac{0.5,0.62}{AH}, \frac{0.5,0.62}{BL}+\frac{0.5,0.62}{BM}, \frac{0,0.25}{CM}+\frac{1,1}{CH}$ |
| 6 | B-3, C-3, D-2 | $\frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0,0.25}{CL}+\frac{1,1}{CM}+\frac{0,0.25}{CH}+\frac{0.5,0.62}{DL}+\frac{0.5,0.62}{DM}$ |
| 7 | C-3, E-2 | $\frac{0,0.25}{CL}+\frac{1,1}{CM}+\frac{0,0.25}{CH}, \frac{0.5,0.62}{EL}+\frac{0.5,0.62}{EM}$ |

For example, the first transaction item A with quantity four generates two fuzzy linguistics terms, AM (A-middle) and AH (A-high), each linguistic term with a fuzzy value of 0.5 and 0.62 as lower and higher values, respectively, as depicted in Table 6.2. Thus, it is challenging to mine MFFIs from two values associated with each fuzzy linguistic term. So, take the fuzzy interval value by taking an average of it and use the centroid type-reduction approach [17] to simplify this complexity. The interval value can be obtained through the application of the following formula.

$$f_{iq1} = \frac{f_{iq1}^{lower} + f_{iq1}^{upper}}{2} \tag{6.1}$$

So, get 0.56 internal fuzzy values of linguistic terms AM and AH according to the given formula. This final transformed first transaction from D is displayed in Table 6.3.

Table 6.3: Final first fuzzy transaction

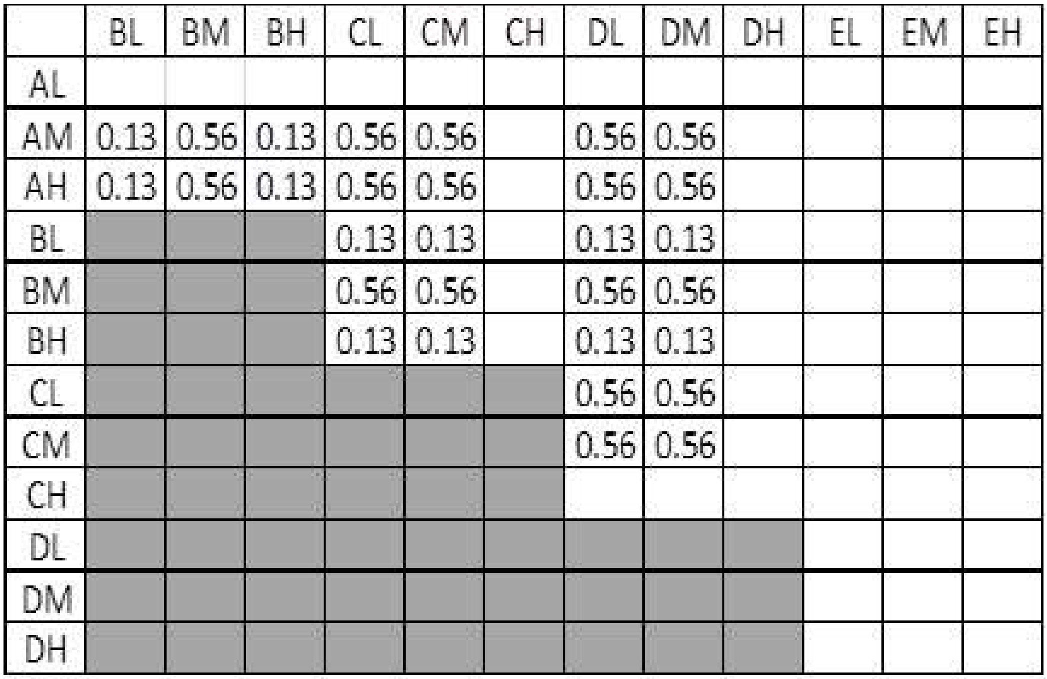


| TID | Final Fuzzy dataset |
|---|---|
| 1 | $\frac{0.56}{AM}+\frac{0.56}{AH}, \frac{0.13}{BL}+\frac{1}{BM}+\frac{0.13}{BH}, \frac{0.56}{CL}+\frac{0.56}{CM}, \frac{0.56}{DL}+\frac{0.56}{DM}$ |

Enhance the adjacency matrix by inputting the minimum fuzzy value for each pair into the respective cell.

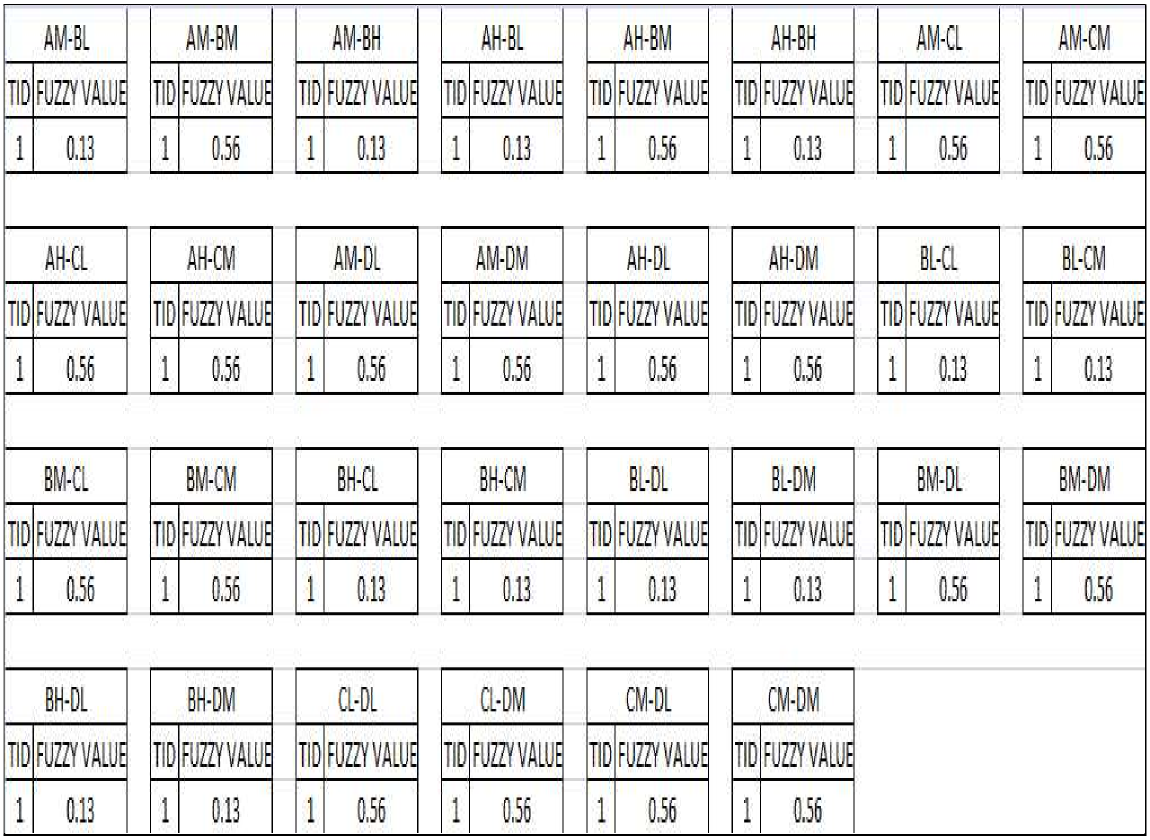
$$\mathrm{AM}(\mathrm{Li},\mathrm{Lj}) = \mathrm{AM}\ (\mathrm{Li},\mathrm{Lj}) + \ \min\ (\mathrm{fwiq},\mathrm{fwjq}) \tag{6.2}$$

fwiq and fwjq are the fuzzy value of the fuzzy items Li and Lj, respectively. Create a Fuzzy-tid-list for Li and Lj if it does not exist. A minimum of fwiq and fwjq was added with transaction ID q to the Fuzzy-tid-list. Completing the first-row adjacency matrix and Fuzzy-tid-list is shown in Figure 6.3 and Figure 6.4, respectively.



| | BL | BM | BH | CL | CM | CH | DL | DM | DH | EL | EM | EH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AL | | | | | | | | | | | | |
| AM | 0.13 | 0.56 | 0.13 | 0.56 | 0.56 | | 0.56 | 0.56 | | | | |
| AH | 0.13 | 0.56 | 0.13 | 0.56 | 0.56 | | 0.56 | 0.56 | | | | |
| BL | | | | 0.13 | 0.13 | | 0.13 | 0.13 | | | | |
| BM | | | | 0.56 | 0.56 | | 0.56 | 0.56 | | | | |
| BH | | | | 0.13 | 0.13 | | 0.13 | 0.13 | | | | |
| CL | | | | | | | 0.56 | 0.56 | | | | |
| CM | | | | | | | 0.56 | 0.56 | | | | |
| CH | | | | | | | | | | | | |
| DL | | | | | | | | | | | | |
| DM | | | | | | | | | | | | |
| DH | | | | | | | | | | | | |

Fig. 6.3: Adjacency Matrix after the first-row scan



| AM-BL | | AM-BM | | AM-BH | | AH-BL | | AH-BM | | AH-BH | | AM-CL | | AM-CM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE |
| 1 | 0.13 | 1 | 0.56 | 1 | 0.13 | 1 | 0.13 | 1 | 0.56 | 1 | 0.13 | 1 | 0.56 | 1 | 0.56 |

| AH-CL | | AH-CM | | AM-DL | | AM-DM | | AH-DL | | AH-DM | | BL-CL | | BL-CM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE |
| 1 | 0.56 | 1 | 0.56 | 1 | 0.56 | 1 | 0.56 | 1 | 0.56 | 1 | 0.56 | 1 | 0.13 | 1 | 0.13 |

| BM-CL | | BM-CM | | BH-CL | | BH-CM | | BL-DL | | BL-DM | | BM-DL | | BM-DM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE |
| 1 | 0.56 | 1 | 0.56 | 1 | 0.13 | 1 | 0.13 | 1 | 0.13 | 1 | 0.13 | 1 | 0.56 | 1 | 0.56 |

| BH-DL | | BH-DM | | CL-DL | | CL-DM | | CM-DL | | CM-DM | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE |
| 1 | 0.13 | 1 | 0.13 | 1 | 0.56 | 1 | 0.56 | 1 | 0.56 | 1 | 0.56 |

Fig. 6.4: Fuzzy-tid-list after the 1st-row scan

The steps to generate an adjacency matrix and a fuzzy-tid-list are shown in Algorithm 6.1 below.

***Algorithm 6.1: Build the AM- Adjacency matrix and FTL- Fuzzy-tid-list***

```
Input: D-Crisp database, M- Total Items
Output: AM and FTL: Fuzzy-tid-list
1: Create AM for m rows and m columns, where m is total fuzzy linguistic terms
2: Set Transaction ID Tid=1
3: From database D, scan line Tq
4: While Tq is in D, proceed to step 9 again
5: Apply the member function type-2 to the quantitative values of each item in  Tq to
    generate the fuzzy linguistic terms fl[] for all items.
     In fl[] contains the average fuzzy value of the linguistic term.
6: Define FV= minimum (FV of fl[i], FV of fl[j]), where FV stand for fuzzy value
    Insert fuzzy values for all co-occurrences of fuzzy linguistic phrases in the adjacency
matrix AM.
```

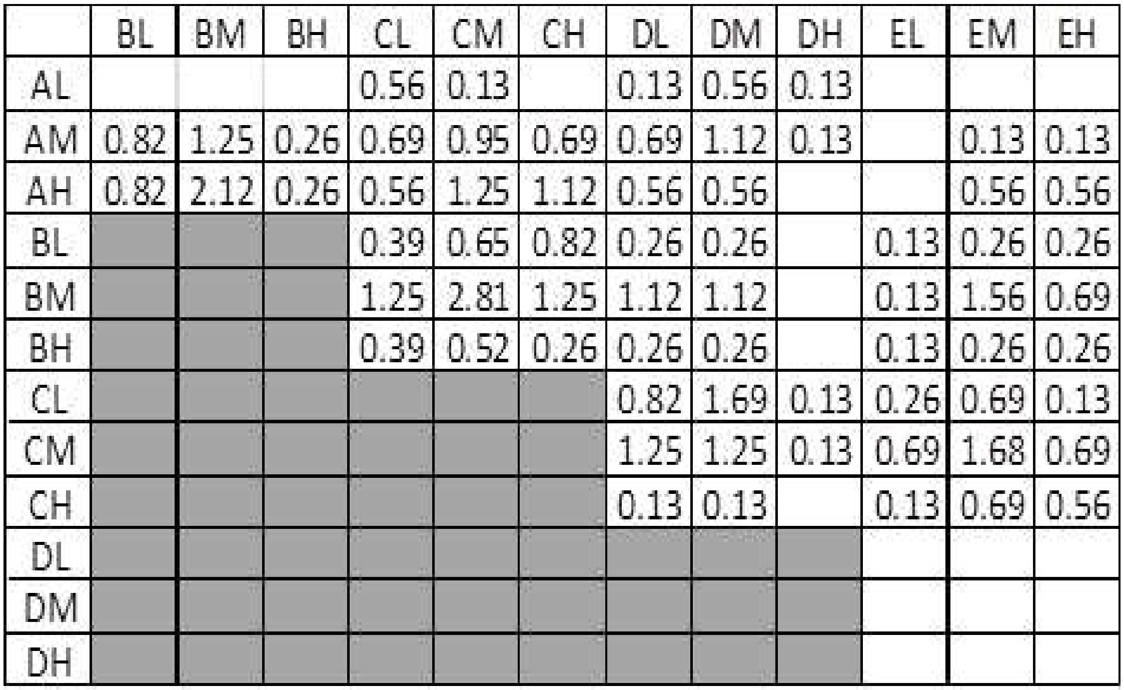
```
Build FTL of “fl[i]- fl[j]“ if not exist.
Build an element using (TID, FV) and add it to the FTL "fl[i]-fl[j]".
Add 1 to Tid and scan the next line from D to Tq.
End.
```

Final transformed all transactions from D into final fuzzy dataset D’ is displayed in Table 6.4 after applying equation 6.1.

Table 6.4: Final fuzzy dataset

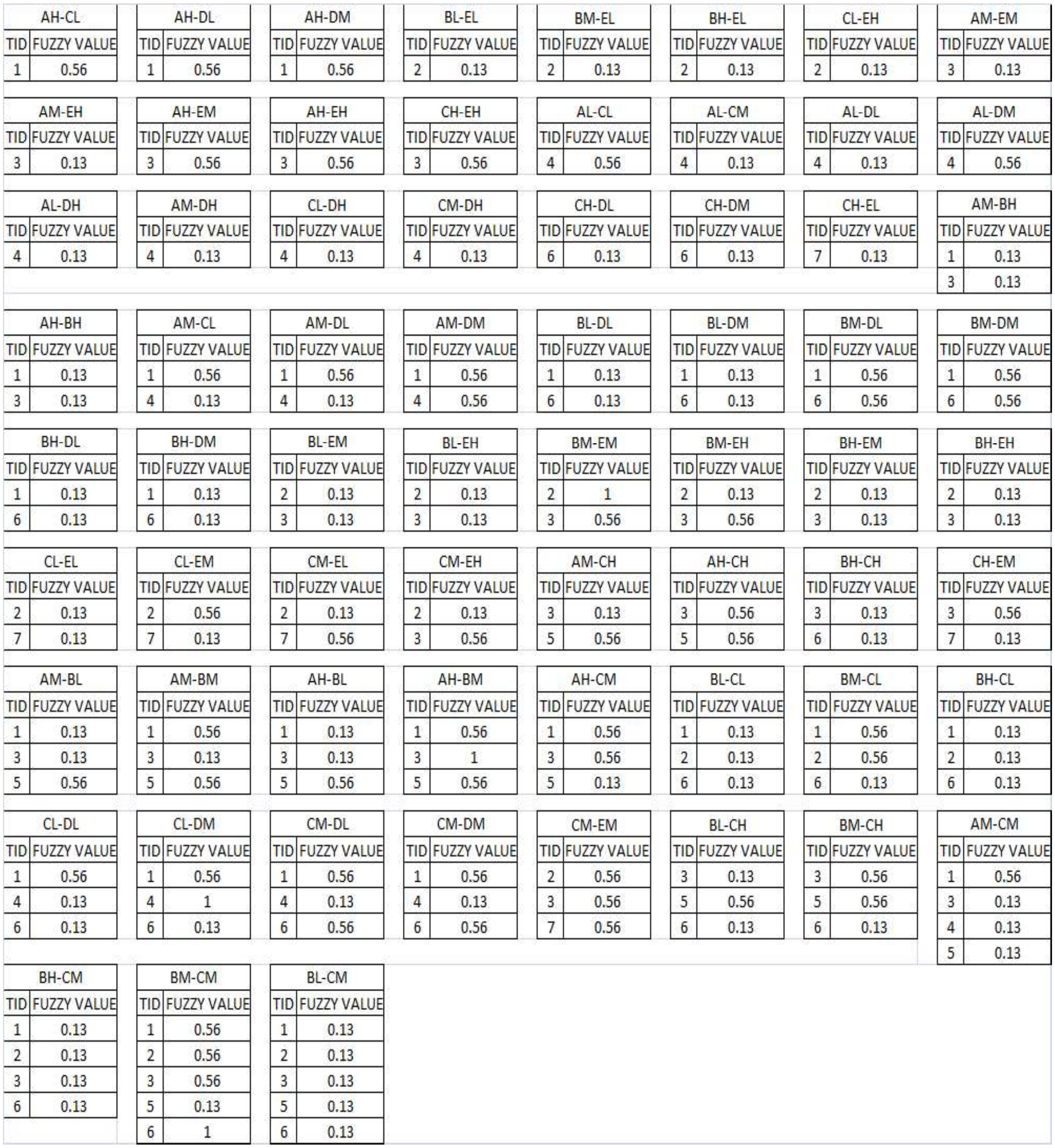
| TID | Final Fuzzy dataset |
|---|---|
| 1 | 0.56/AM + 0.56/AH , 0.13/BL + 1/BM + 0.13/BH , 0.56/CL + 0.56/CM , 0.56/DL + 0.56/DM |
| 2 | 0.13/BL + 1/BM + 0.13/BH , 0.56/CL + 0.56/CM , 0.13/EL + 1/EM + 0.13/EH |
| 3 | 0.13/AM + 1/AH , 0.13/BL + 1/BM + 0.13/BH , 0.56/CM + 0.56/CH , 0.56/EM + 0.56/EH |
| 4 | 0.56/AL + 0.56/AM , 1/CL + 0.13/CM , 0.13/DL + 1/DM + 0.13/DH |
| 5 | 0.56/AM + 0.56/AH , 0.56/BL + 0.56/BM , 0.13/CM + 1/CH |
| 6 | 0.13/BL + 1/BM + 0.13/BH , 0.13/CL + 1/CM + 0.13/CH , 0.56/DL + 0.56/DM |
| 7 | 0.13/CL + 1/CM + 0.13/CH , 0.56/EL + 0.56/EH |

As per the algorithm, after reading all rows, the final adjacency matrix and fuzzy-tid-list list are shown in Figures 6.5 and 6.6, respectively.



|  | BL | BM | BH | CL | CM | CH | DL | DM | DH | EL | EM | EH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AL |  |  |  | 0.56 | 0.13 |  | 0.13 | 0.56 | 0.13 |  |  |  |
| AM | 0.82 | 1.25 | 0.26 | 0.69 | 0.95 | 0.69 | 0.69 | 1.12 | 0.13 |  | 0.13 | 0.13 |
| AH | 0.82 | 2.12 | 0.26 | 0.56 | 1.25 | 1.12 | 0.56 | 0.56 |  |  | 0.56 | 0.56 |
| BL | ▨ | ▨ | ▨ | 0.39 | 0.65 | 0.82 | 0.26 | 0.26 |  | 0.13 | 0.26 | 0.26 |
| BM | ▨ | ▨ | ▨ | 1.25 | 2.81 | 1.25 | 1.12 | 1.12 |  | 0.13 | 1.56 | 0.69 |
| BH | ▨ | ▨ | ▨ | 0.39 | 0.52 | 0.26 | 0.26 | 0.26 |  | 0.13 | 0.26 | 0.26 |
| CL | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | 0.82 | 1.69 | 0.13 | 0.26 | 0.69 | 0.13 |
| CM | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | 1.25 | 1.25 | 0.13 | 0.69 | 1.68 | 0.69 |
| CH | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | 0.13 | 0.13 |  | 0.13 | 0.69 | 0.56 |
| DL | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |  |  |  |
| DM | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |  |  |  |
| DH | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |  |  |  |

Fig. 6.5: Adjacency Matrix after all row scan



| AH-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |

| AH-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |

| AH-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |

| BL-EL | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |

| BM-EL | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |

| BH-EL | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |

| CL-EH | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |

| AM-EM | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.13 |

| AM-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.13 |

| AH-EM | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.56 |

| AH-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.56 |

| CH-EH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.56 |

| AL-CL | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.56 |

| AL-CM | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.13 |

| AL-DL | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.13 |

| AL-DM | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.56 |

| AL-DH | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.13 |

| AM-DH | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.13 |

| CL-DH | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.13 |

| CM-DH | |
|---|---|
| TID | FUZZY VALUE |
| 4 | 0.13 |

| CH-DL | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.13 |

| CH-DM | |
|---|---|
| TID | FUZZY VALUE |
| 6 | 0.13 |

| CH-EL | |
|---|---|
| TID | FUZZY VALUE |
| 7 | 0.13 |

| AM-BH | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 3 | 0.13 |

| AH-BH | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 3 | 0.13 |

| AM-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 0.13 |

| AM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 0.13 |

| AM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 0.56 |

| BL-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 6 | 0.13 |

| BL-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 6 | 0.13 |

| BM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 6 | 0.56 |

| BM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 6 | 0.56 |

| BH-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 6 | 0.13 |

| BH-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 6 | 0.13 |

| BL-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 3 | 0.13 |

| BL-EH | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 3 | 0.13 |

| BM-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 1 |
| 3 | 0.56 |

| BM-EH | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 3 | 0.56 |

| BH-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 3 | 0.13 |

| BH-EH | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 3 | 0.13 |

| CL-EL | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 7 | 0.13 |

| CL-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.56 |
| 7 | 0.13 |

| CM-EL | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 7 | 0.56 |

| CM-EH | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.13 |
| 3 | 0.56 |

| AM-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.13 |
| 5 | 0.56 |

| AH-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.56 |
| 5 | 0.56 |

| BH-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.13 |
| 6 | 0.13 |

| CH-EM | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.56 |
| 7 | 0.13 |

| AM-BL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 3 | 0.13 |
| 5 | 0.56 |

| AM-BM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 3 | 0.13 |
| 5 | 0.56 |

| AH-BL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 3 | 0.13 |
| 5 | 0.56 |

| AH-BM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 3 | 1 |
| 5 | 0.56 |

| AH-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 3 | 0.56 |
| 5 | 0.13 |

| BL-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 2 | 0.13 |
| 6 | 0.13 |

| BM-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 2 | 0.56 |
| 6 | 0.13 |

| BH-CL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 2 | 0.13 |
| 6 | 0.13 |

| CL-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 0.13 |
| 6 | 0.13 |

| CL-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 1 |
| 6 | 0.13 |

| CM-DL | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 0.13 |
| 6 | 0.56 |

| CM-DM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 4 | 0.13 |
| 6 | 0.56 |

| CM-EM | |
|---|---|
| TID | FUZZY VALUE |
| 2 | 0.56 |
| 3 | 0.56 |
| 7 | 0.56 |

| BL-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.13 |
| 5 | 0.56 |
| 6 | 0.13 |

| BM-CH | |
|---|---|
| TID | FUZZY VALUE |
| 3 | 0.56 |
| 5 | 0.56 |
| 6 | 0.13 |

| AM-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 3 | 0.13 |
| 4 | 0.13 |
| 5 | 0.13 |

| BH-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 2 | 0.13 |
| 3 | 0.13 |
| 6 | 0.13 |

| BM-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.56 |
| 2 | 0.56 |
| 3 | 0.56 |
| 5 | 0.13 |
| 6 | 1 |

| BL-CM | |
|---|---|
| TID | FUZZY VALUE |
| 1 | 0.13 |
| 2 | 0.13 |
| 3 | 0.13 |
| 5 | 0.13 |
| 6 | 0.13 |

Fig. 6.6: Fuzzy-tid-list after all row scans from dataset D

## 6.2 From Adjacency matrix Mining MFFIs using MFFPA-2 method

In this phase extraction of MFFIs using the adjacency matrix (M), and Fuzzy-tid-list constructed from the result generated in first phase. Select the cell with a value greater than or equal to min-support in the row of Adjacency matrix M. They declared the identified cell row-column combination as fuzzy 2-frequent itemsets ($FL_2$) and fetched

Fuzzy-tid-list for the identified cell. For the example in the first row in Figure 6.5 with headed AL, there is no cell with value >= $\delta$; here, min-support $\delta$=1. Scan the second row in which two cells, namely AM-BM and AM-DM, find which satisfies min-support threshold $\delta$ so fetched Fuzzy-tid-list of AM-BM and AM-DM. Next, recursively create fuzzy k-frequent itemsets, such as $FL_k$ (K>2), in a subsequent step by intersection-operating TIDs on $FL_{k-1}$. The binary search method can be used to find combined fast fuzzy lists. To create the Fuzzy-tid-list for k-frequent itemsets (k>2), the existing $FL_{k-1}$ Fuzzy-tid-list are combined. Elements in a newly created Fuzzy-tid-list are those with a common Tid in an existing Fuzzy-tid-list.

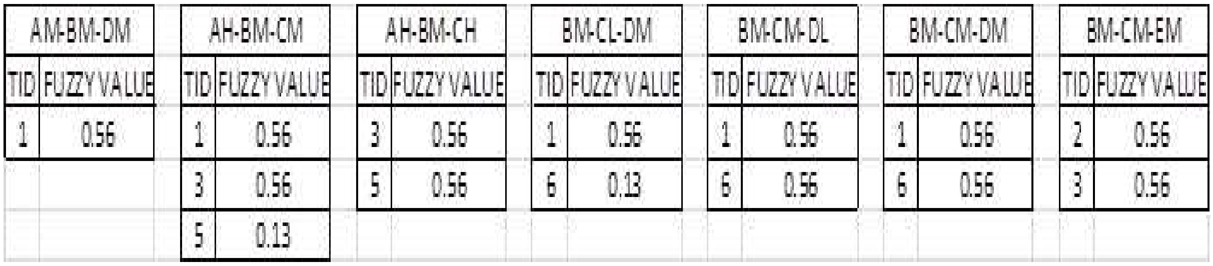
Only joining fuzzy itemsets that can create their superset directly identified from the adjacency matrix M will reduce the search space and candidate set. As found, $FL_2$ is from the second row AM-BM and AM-DM, so the subsequent possible superset is AM-BM-DM. If the BM-DM value from the BM row and DM column cell value satisfy the min-support threshold, generate the AM-BM-DM Fuzzy-tid-list by joining the AM-BM Fuzzy-tid-list and AM-DM Fuzzy-tid-list using the intersection operation on it. In the fifth row headed by BM, five cells satisfy $\delta$, so generated $FL_2$ from this row is BM-CL, BM-CM, BM-CH, BM-DL, and BM-DM. Next subsequent possible $FL_3$ are BM-CL-DL, BM-CL-DM, BM-CM-DL, BM-CM-DM, BM-CH-DL and BM-CH-DM. BM-CL-DL does not join because of CL-DL value, which does not satisfy $\delta$, so ignore this set directly without generating its candidate itemsets or not join BM-CL-DL. This way drastically joins operation minimize or candidate itemsets, improving running time efficiency, for this MFFPA-2 method is shown in algorithms 6.2 and 6.3.

***Algorithm 6.2: MFFPA-2***

```
Input: min_supp, FTLs: Fuzzy-tid-list, AM: Adjacency Matrix
1: Do for every row in AM
FTL ← null;   \\ Fuzzy-tid-list Initialization
2:  for each row's cell
            If cell value >= min_supp
            Get fuzzy-tid-list of ([row.id]-[col.id]) from FTLs into FTL
            Call MFFPA-2-miner with FTL
```

```
Algorithm 6.3: MFFPA-2-miner
Input: min_supp, FTL: Fuzzy-tid-list, AM: Adjacency Matrix
Output: MFFI,
do for every fuzzy-list P in FTL
    MFFIs ← P ∪MFFIs.
    TFL ← null; //temporary Fuzzy-tid-list
    do for every fuzzy-tid-list Q after P in FTL
        If P.product = Q.product, then
            skip
        else If M[P.product][Q.product] >= min_supp then
    Join P, Q and insert into TFL;
    MFFPA-2-Miner(TFL);
Return MFFIs.
```

After reviewing each row by the algorithm-derived candidate Fuzzy-tid-list depicted in Figure 6.7, numerous potential candidates emerge. However, this method produces only seven Tid-fuzzy sets from these, constituting five fuzzy frequent itemsets with a length of 3. Next, it does not generate a candidate set for length 4, which is known directly from the adjacency matrix. So, using the MFPPA-2 method and Adjacency matrix generates fewer candidate sets than the state-of-art method.



| AM-BM-DM | | AH-BM-CM | | AH-BM-CH | | BM-CL-DM | | BM-CM-DL | | BM-CM-DM | | BM-CM-EM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE | TID | FUZZY VALUE |
| 1 | 0.56 | 1 | 0.56 | 3 | 0.56 | 1 | 0.56 | 1 | 0.56 | 1 | 0.56 | 2 | 0.56 |
| | | 3 | 0.56 | 5 | 0.56 | 6 | 0.13 | 6 | 0.56 | 6 | 0.56 | 3 | 0.56 |
| | | 5 | 0.13 | | | | | | | | | | |

Fig. 6.7: Fuzzy-tid-list after all row scans from Matrix M

From Figure 6.7, the MFFPA-2 technique generates only 7 joins, of which 5 are multiple fuzzy frequent itemsets, so the MFFPA-2 approach generates only 2 unnecessary candidate itemsets. Generated 3-MFFIs are: AH-BM-CM, AH-BM-CH, BM-CM-DL, BM-CM-DM, AND BM-CM-EM. Extensions are discarded before joining since they are not fuzzy frequent itemsets directly from the adjacency matrix. This way, the join operation minimizes the candidate set, improving the running time performance.

The AMFFI technique in Chapter 4 generates 3-MFFIs for the discussed example, which is only 2, but as the type-2 membership function in the MFFPA-2 technique generates 3-MFFIs, it is 5.

The subsequent chapter discusses performance comparisons of the MFFPA-2 technique with state-of-the-art methods for running time, memory utilization, and node join counts.

# CHAPTER 7

# Experimental Study and Performance Evaluation of MFFPA-2

Here, we contrast the MFFPA-2 achievement in the recommended method with the list-based techniques by Lin et al. [61] and EFM [14]. We Implement the proposed MFFPA-2, EFM, and Lin's method in Java. The outcomes are examined using two standard datasets, chess and mushroom [62], and one artificial T10I4D100k dataset [62]. Details of datasets are given in Table 7.1 below. The quantities of objects in the datasets are provided at random intervals between 1 and 7. The outcome of the experiment runtime, join count, and memory usage were all examined.

Table 7.1: Details of datasets

| Sr. No. | Dataset | No. Of. Transactions | No. of Items |
|---|---|---|---|
| 1 | Chess | 3196 | 75 |
| 2 | Mushroom | 8416 | 119 |
| 3 | T10I4D100k | 100000 | 1000 |

## 7.1 Runtime Analysis

We implemented MFFPA-2, EFM [14], and Lin's [61] methods using the type-2 member function with 3-term fuzzy linguistic terms. To assess the execution time performance of the applied techniques, various minimum support threshold values were employed. Figures 7.1 through 7.3 show the findings of the execution running time evaluation on the chess dataset, mushroom dataset, and T10I4D100k dataset.

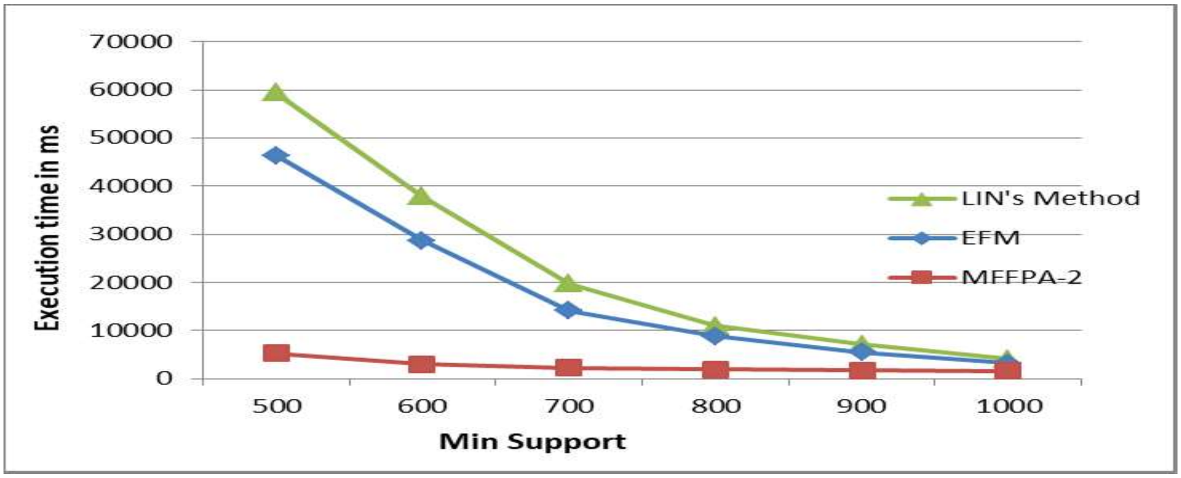


Fig. 7.1: Comparisons of execution times: Chess dataset

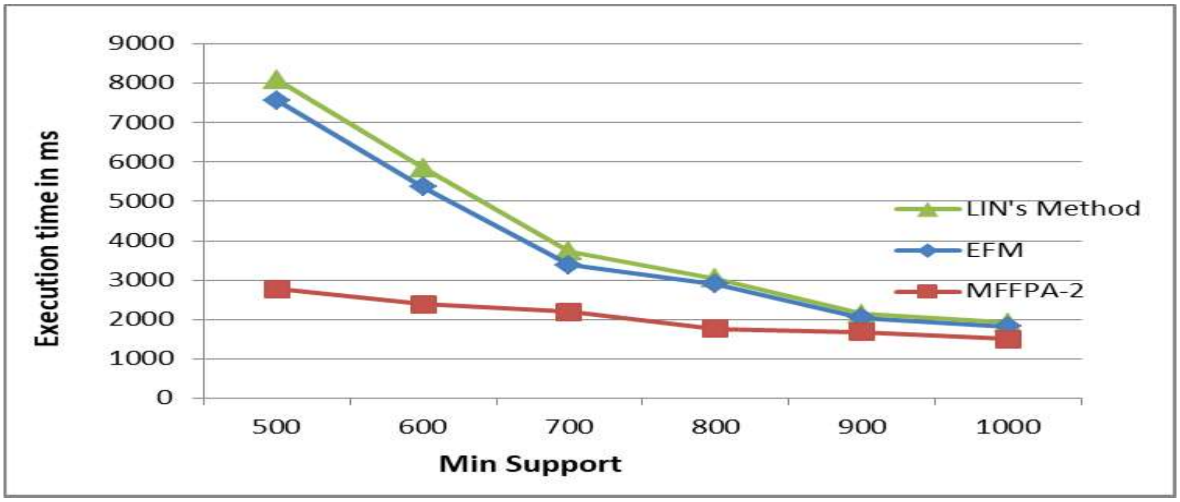


Fig. 7.2: Comparisons of execution times: Mushroom dataset

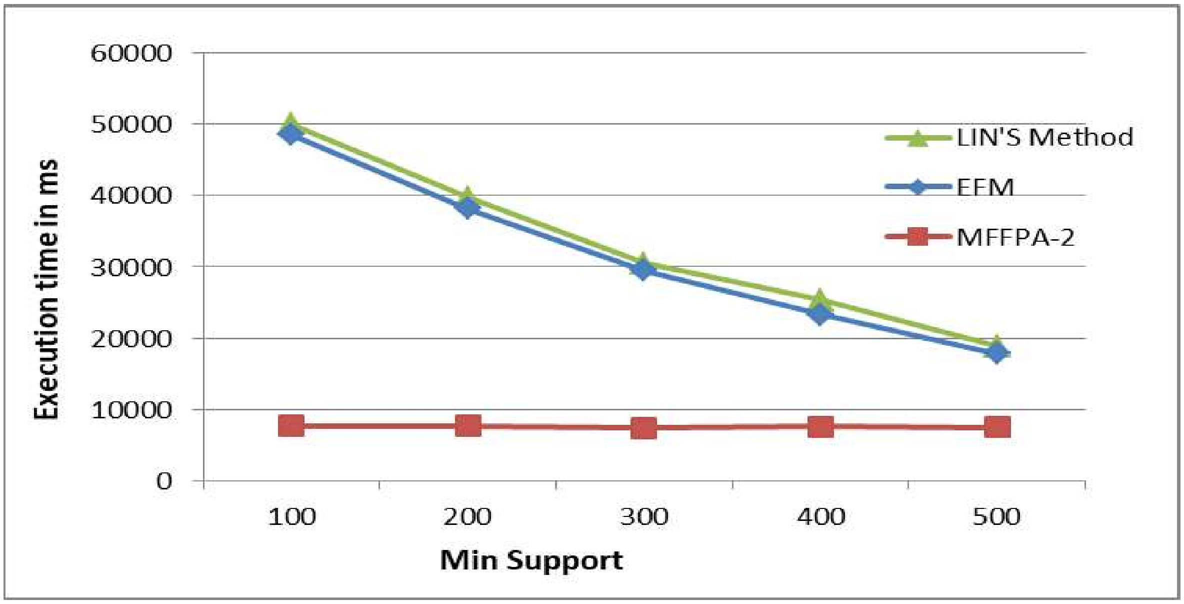


Fig.7.3: Comparisons of execution times: T10I4D100k dataset

From the results, it is observed that the proposed MFFPA-2 approach works better than the alternative method. A reduced minimum support criterion also shows how resilient the MFFPA-2 technique is.

### 7.2 The number of Join Counts Analysis

The number of joins made during the formation of MFFIs is considered while evaluating performance in this area. Figures 7.4 through 7.6 show the findings of the evaluations of the number of join counts on the chess, mushroom, and T10I4D100k datasets.

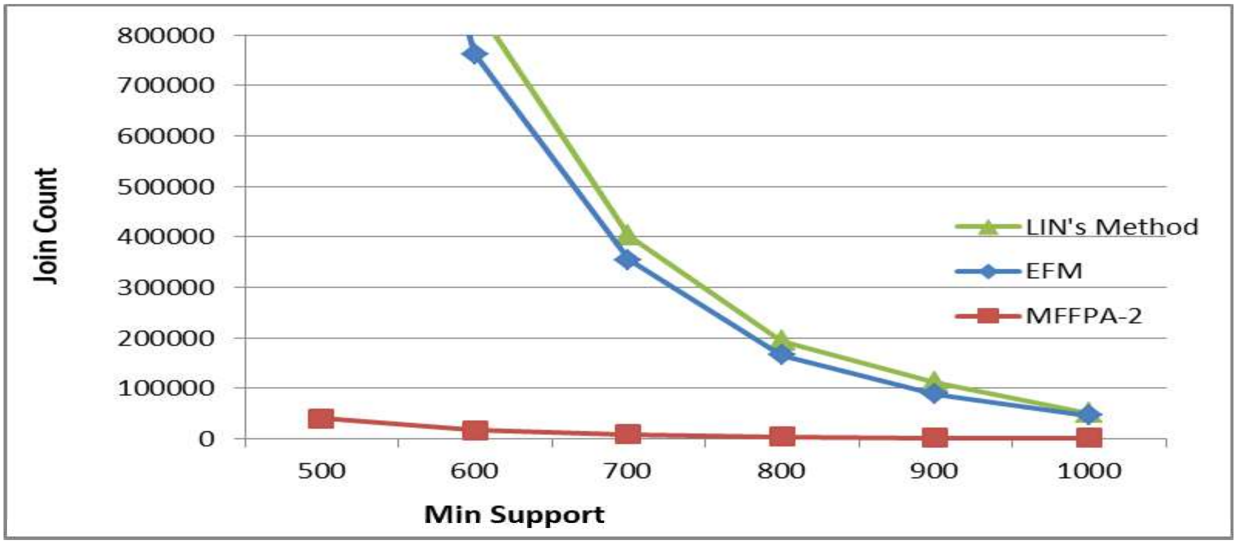


Fig. 7.4: Comparisons of Join Counts: Chess dataset

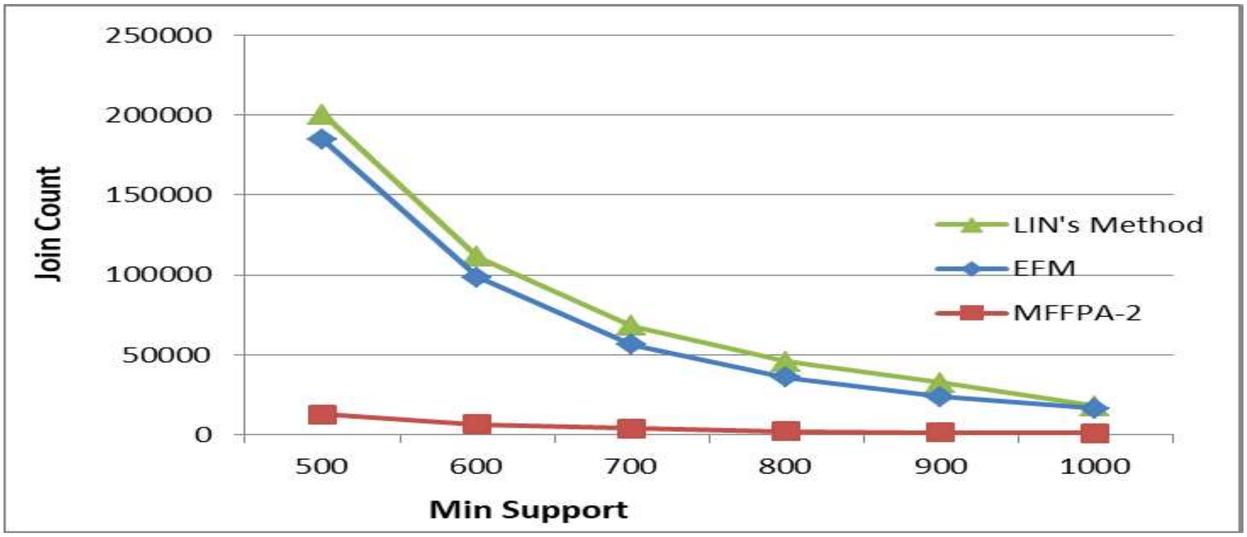


Fig. 7.5: Comparisons of Join Counts: Mushroom dataset

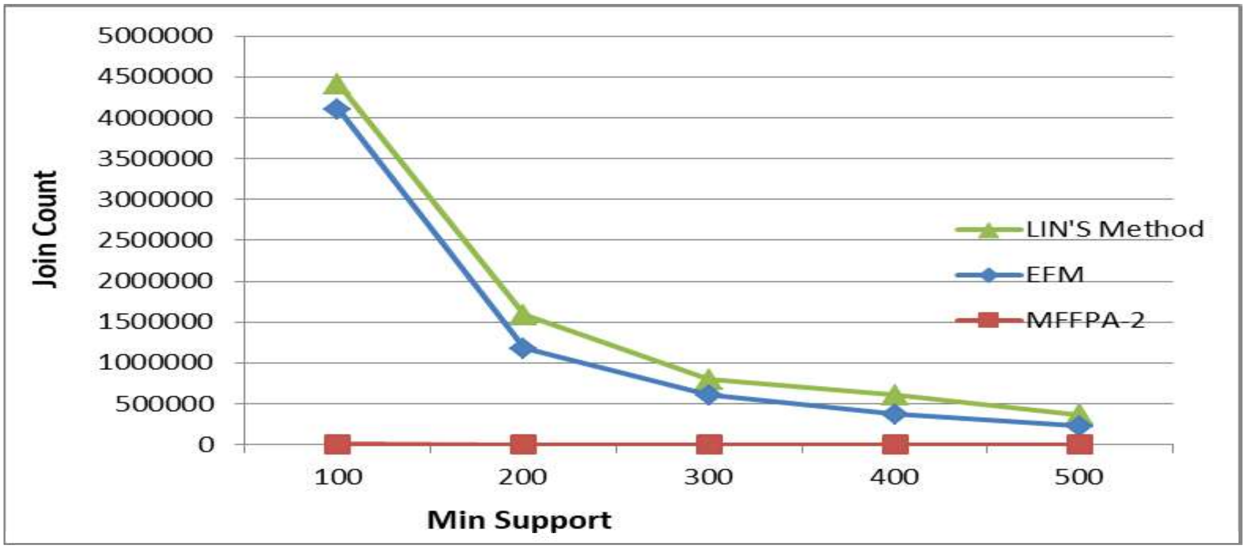


Fig. 7.6: Comparisons of Join Counts: T10I4D100k dataset

The findings indicate that MFFPA-2 generates fewer join counts (candidate itemsets). It was noted that the MFFPA-2 method's join count performance is by far the most impressive. The proposed MFFPA-2 method produces fewer candidate itemsets than cutting-edge techniques.

### 7.3 Memory Utilization Analysis

In the conducted studies, the efficiency assessment is based on the degree to which memory was utilized. Figures 7.7 through 7.9 display the outcomes of memory utilization on the chess, mushroom, and T10I4D100k datasets.

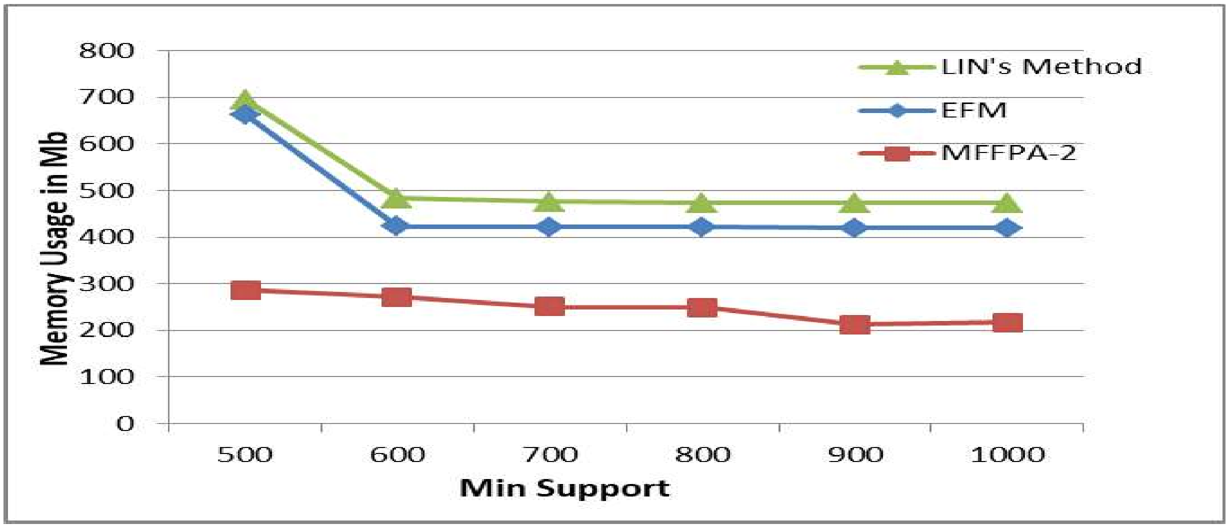


Fig. 7.7: Comparisons of Memory Usage: Chess dataset

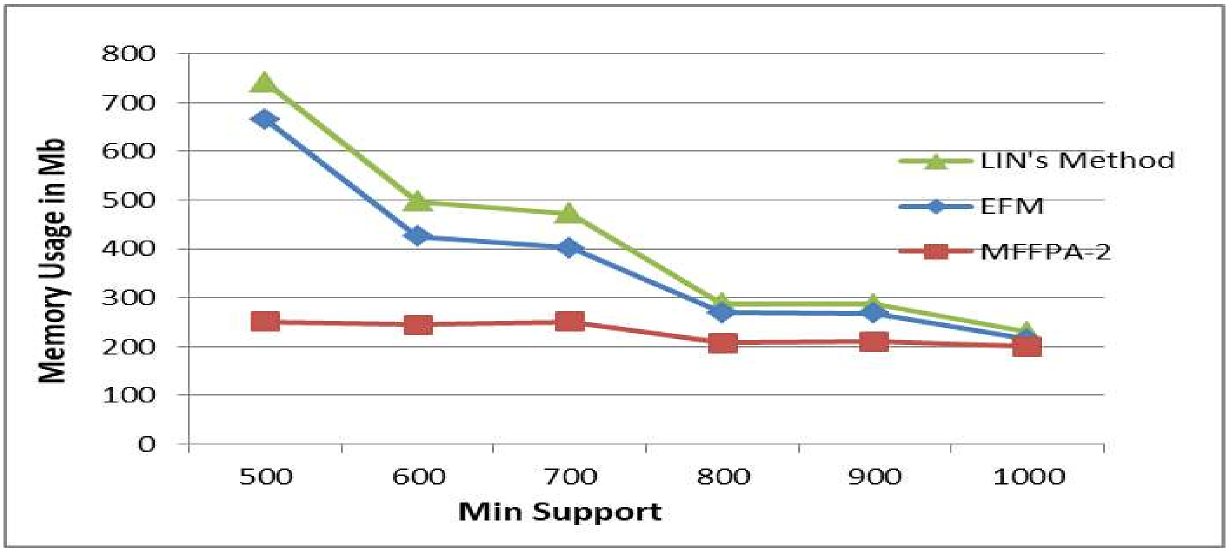


Fig. 7.8: Comparisons of Memory Usage: Mushroom dataset

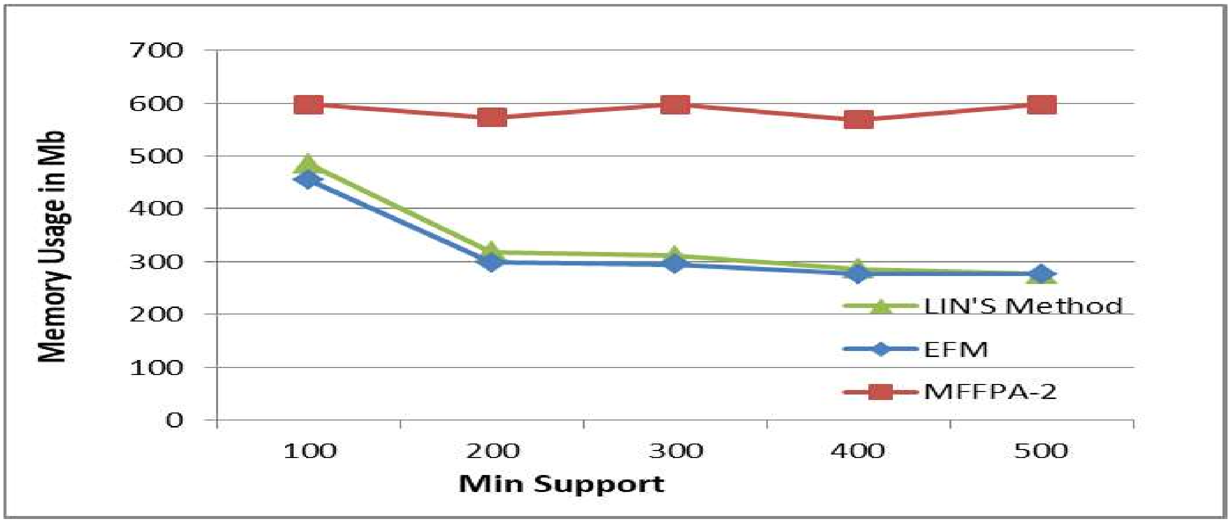


Fig. 7.9: Comparisons of Memory Usage: T10I4D100k dataset

The results show that on the chess and mushroom datasets, the MFFPA-2 method utilizes less memory than the comparison strategy. It was noted that the MFFPA-2 method uses more memory than the compared approach on the artificial T10I4D100k dataset. We may deduce from additional trials with other datasets that in a particular case where a dataset has more than 1000 items, the suggested MFFPA-2 will need a more significant memory.

## 7.4 Overall improvement in percentage

The proposed MFFPA-2, Lin's [61], and EFM [14] methods were evaluated with different min-support thresholds to compare execution running time, number of join counts, and memory utilization. The overall improvement of MFFPA-2 vs. EFM in percentage with different min-support on chess, mushroom, and T10I4D100k datasets are shown in Table 7.2, Table 7.3, and Table 7.4, respectively.

Table 7.2: Improvement MFFPA-2 vs. EFM on the chess dataset

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 500 | 97.79 | 88.61 | 56.86 |
| 600 | 97.80 | 89.34 | 35.70 |
| 700 | 97.64 | 83.88 | 40.52 |
| 800 | 97.37 | 77.27 | 41.00 |
| 900 | 97.34 | 67.25 | 49.16 |
| 1000 | 97.09 | 48.28 | 48.10 |

Table 7.3: Improvement MFFPA-2 vs. EFM on Mushroom dataset

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 500 | 93.15 | 63.27 | 62.26 |
| 600 | 93.50 | 55.53 | 42.35 |
| 700 | 93.39 | 35.47 | 37.56 |
| 800 | 93.96 | 39.14 | 23.05 |
| 900 | 94.61 | 17.61 | 21.64 |
| 1000 | 94.91 | 17.03 | 06.98 |

Table 7.4: Improvement MFFPA-2 vs EFM on T10I4D100k dataset

| Min Support | Improvement in % | | |
|---|---|---|---|
| | Join count | Execution time | Memory Usage |
| 100 | 99.90 | 84.21 | -31.72 |
| 200 | 99.98 | 79.86 | -92.28 |
| 300 | 99.99 | 75.04 | -102.71 |
| 400 | 100.0 | 67.60 | -106.16 |
| 500 | 100.0 | 58.16 | -116.67 |

The above tables 7.2 to 7.4 demonstrate that the MFFPA-2 approach is better than the current EFM approach concerning execution time and node join counts.

# CHAPTER 8
# Conclusion and Future Enhancement

## 8.1 Conclusions

Frequent itemsets mining is essential in this era of growing e-commerce business. Enhancing the efficiency of mining frequent itemsets can be achieved through the application of fuzzy theory. The enhancement of execution speed and reduction in memory usage for mining Fuzzy Frequent itemsets can be achieved by minimizing both the candidate itemsets and the frequency of database scans. An efficient and optimal mining method is desirable with the growing demand for identifying frequent itemsets. The proposed adjacency matrix-based Fuzzy frequent itemsets mining approach significantly reduces the candidate itemsets and scans the database only once. Hence, the proposed approach improves performance. Proposed approaches, AMFFI and MFFPA-2, efficiently work in fuzzy frequent itemtsets mining.

Experimental analysis shows improvement of AMFFI Vs. MFFI-miner and MFFPA-2 Vs. EFM for different databases with different minimum support. The outcomes of the suggested methods exhibit superior performance concerning execution time and the number of join counts. However, they require more memory when the number of items increases.

The following tables summarize the average improvement by proposed method AMFFI over MFFI-miner and MFFPA-2 over EFM for different min support. Table 8.1 shows improvement for the 3-term member function of AMFFI Vs. MFFI-miner, Table 8.2 shows improvement for the 5-term member function of AMFFI Vs. MFFI-miner and Table 8.3 show improvement of MFFPA-2 Vs. EFM.

Table 8.1: Average improvement for a 3-term member function of AMFFI Vs. MFFI-miner

| Database | Join count | Execution time | Memory Usage |
|---|---|---|---|
| Chess | 97.50 | 81.47 | 45.22 |
| Mushroom | 93.92 | 42.07 | 32.84 |
| T10I4D100K | 99.97 | 72.97 | -89.91 |

Table 8.2: Average improvement for a 5-term member function of AMFFI Vs. MFFI-miner

| Database | Join count | Execution time | Memory Usage |
|---|---|---|---|
| Chess | 96.75 | 14.61 | 32.08 |
| Mushroom | 73.01 | 8.68 | 14.45 |
| Chicago_crime_2001-2017 | 70.62 | 53.67 | 26.19 |

Table 8.3: Average improvement of MFFPA-2 vs. EFM

| Database | Join count | Execution time | Memory Usage |
|---|---|---|---|
| Chess | 97.50 | 75.77 | 45.22 |
| Mushroom | 93.92 | 38.01 | 32.31 |
| T10I4D100K | 99.97 | 72.97 | -89.91 |

Proposed algorithm shows an average 8% to 81% improvement in execution time compared to existing state-of-the-art methods for various datasets. The AMFFI improves execution time by 8% to 81% and node join count by 93% to 99%. The MFFPA-2 improves execution time by 38% to 75% and node join count by 93% to 99%.

## 8.2 Future Enhancement

The work is limited to the static database. The database is continuously updated in real-life applications, so the work will also expand for the stream and dynamic datasets.

In the future, the adjacency matrix structure will be enhanced to minimize memory requirements. For joining two fuzzy-tid-lists in place of binary search technique different search exploration method can be used to minimize searching time in fuzzy-tid-list (node) join operation.

# References

[1] Sadiku, A. E. M., & Matthew, N. O. (2015) Shadare and SM, “A Brief Introduction to Data Mining,”. Eur. Sci. J, 11(21), pp. 1-3.

[2] Xu JJ. (2014). Knowledge discovery and data mining. in Computing Handbook Information Systems and Information Technology, Third Ed (pp. 1–22).

[3] Maimon, O., Rokach, L. (2009). Introduction to Knowledge Discovery and Data Mining. In: Maimon, O., Rokach, L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA.

[4] Agrawal, R., Imielinski, T., & Swami, A. (1993). Database mining: A performance perspective. IEEE transactions on knowledge and data engineering, 5(6), 914-925.

[5] Kumbhare, T. A., & Chobe, S. V. (2014). An overview of association rule mining algorithms. International Journal of Computer Science and Information Technologies, 5(1), 927-930.

[6] Berkhin, P. (2006). A survey of clustering data mining techniques. In Grouping multidimensional data: Recent advances in clustering (pp. 25-71). Berlin, Heidelberg: Springer Berlin Heidelberg.

[7] Antonelli, M., Ducange, P., Marcelloni, F., & Segatori, A. (2015). A novel associative classification model based on a fuzzy frequent pattern mining algorithm. Expert Systems with Applications, 42(4), 2086-2097.

[8] Hu, K., Lu, Y., Zhou, L., & Shi, C. (1999). Integrating classification and association rule mining: A concept lattice framework. In New Directions in Rough Sets, Data Mining, and Granular-Soft Computing: 7th International Workshop, RSFDGrC’99, Yamaguchi, Japan, November 9-11, 1999. Proceedings 7 (pp. 443-447). Springer Berlin Heidelberg.

[9] Min, H. (2006). Developing the profiles of supermarket customers through data mining. The Service Industries Journal, 26(7), 747-763.

[10] Luna, J. M., Fournier-Viger, P., & Ventura, S. (2019). Frequent itemset mining: A 25-year review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(6), e1329.

[11] Brauckhoff, D., Dimitropoulos, X., Wagner, A., & Salamatian, K. (2009, November). Anomaly extraction in backbone networks using association rules.

In Proceedings of the 9th ACM SIGCOMM conference on Internet measurement (pp. 28-34).

[12] Hong, T. P., Kuo, C. S., & Chi, S. C. (1999). Mining association rules from quantitative data. Intelligent data analysis, 3(5), 363-376.

[13] Lin, J. C. W., Li, T., Fournier-Viger, P., Hong, T. P., Wu, J. M. T., & Zhan, J. (2017). Efficient mining of multiple fuzzy frequent itemsets. International Journal of Fuzzy Systems, 19, 1032-1040..

[14] Lin, J. C. W., Wu, J. M. T., Djenouri, Y., Srivastava, G., & Hong, T. P. (2020). Mining multiple fuzzy frequent patterns with compressed list structures. In 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1-8).

[15] Ristovska, K., & Ristovska, A. (2014). The impact of globalization on the business. Economic Analysis, 47(3-4), 83-89.

[16] Voleti S (2017). The Value of Data : A Motivating Example. Essentials Bus. Anal. pp. 19–39.

[17] Bavdaž, M., Snijkers, G., Sakshaug, J. W., Brand, T., Haraldsen, G., Kurban, B., ... & Willimack, D. K. (2020). Business data collection methodology: Current state and future outlook. Statistical Journal of the IAOS, 36(3), 741-756.

[18] Chan, S. L., & Ip, W. H. (2011). A dynamic decision support system to predict the value of customers for new product development. Decision support systems, 52(1), 178-188.

[19] Han, J., Kamber, M., & Pei, J. (2012). Data mining concepts and techniques third edition. University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University.

[20] Tan, P. N., Steinbach, M., & Kumar, V. (2016). Introduction to data mining. Pearson Education India.

[21] Kesavaraj, G., & Sukumaran, D.S. (2013). A study on classification techniques in data mining. 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 1-7

[22] Pande, S., Sambare, S.S., & Thakre, V.M. (2012). Data Clustering Using Data Mining Techniques.

[23] Abhinav Rai (2022). An Overview of Association Rule Mining & its Applications;5:927–30.

[24] Borgelt, C. (2012). Frequent item set mining. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2(6), 437-456.

[25] Han, J., Kamber, M., & Pei, J. (2012). Data mining concepts and techniques third edition. University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University.

[26] Grahne, G., & Zhu, J. (2005). Fast algorithms for frequent itemset mining using fp-trees. IEEE transactions on knowledge and data engineering, 17(10), 1347-1362.

[27] Rinaldi, M., Parretti, C., Salimbeni, L. B., & Citti, P. (2015). Conceptual design of a decision support system for the economic sustainability of nonprofit organizations. Procedia CIRP, 34, 119-124.

[28] Pyun, G., Yun, U., & Ryu, K. H. (2014). Efficient frequent pattern mining based on linear prefix tree. Knowledge-Based Systems, 55, 125-139.

[29] Zhang, X. H., He, Y. D., Wan, J. H., & Zhao, H. (2001). An Improved Algorithm for Mining Association Rules. JOURNAL-NORTHEASTERN UNIVERSITY NATURAL SCIENCE, 22, 401-404.

[30] Han, J., Pei, J., & Tong, H. (2022). Data mining: concepts and techniques. Morgan Kaufmann.

[31] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In Proc. 20th int. conf. Very large databases, VLDB (Vol. 1215, pp. 487-499).

[32] J. Friedman and L. A. Z. Fuzzy (1970), "Similarity relations and fuzzy orderings," North-Holland Publishing Company.

[33] Chan, K. C., & Au, W. H. (1997). Mining fuzzy association rules. In Proceedings of the Sixth International Conference on Information and Knowledge Management (pp. 209-215).

[34] Kuok, C. M., Fu, A., & Wong, M. H. (1998). Mining fuzzy association rules in databases. ACM Sigmod Record, 27(1), 41-46.

[35] Hong, T. P., Kuo, C. S., & Wang, S. L. (2004). A fuzzy AprioriTid mining algorithm with reduced computational time. Applied Soft Computing, 5(1), 1-10.

[36] Delgado, M., Marín, N., Sánchez, D., & Vila, M. A. (2003). Fuzzy association rules: general model and applications. IEEE Transactions on Fuzzy Systems, 11(2), 214-225.

[37] Papadimitriou, S., & Mavroudi, S. (2005). The fuzzy frequent pattern tree. In The WSEAS International Conference on Computers (pp. 1-7).

[38] Hong, T. P., Lin, C. W., & Wu, Y. L. (2008). Incrementally fast updated frequent pattern trees. Expert Systems with Applications, 34(4), 2424-2435.

[39] Lin, C. W., Hong, T. P., & Lu, W. H. (2010). Linguistic data mining with fuzzy FP-trees. Expert Systems with Applications, 37(6), 4560-4567.

[40] Lin, C. W., Hong, T. P., & Lu, W. H. (2010). An efficient tree-based fuzzy data mining approach. International Journal of Fuzzy Systems, 12(2), 150-157.

[41] Lin, C. W., & Hong, T. P. (2014). Mining fuzzy frequent itemsets based on UBFFP trees. Journal of Intelligent & Fuzzy Systems, 27(1), 535-548.

[42] Mishra, S., Mishra, D., & Satapathy, S. K. (2011, April). Fuzzy pattern tree approach for mining frequent patterns from gene expression data. In 2011, the 3rd International Conference on Electronics Computer Technology (Vol. 2, pp. 359-363).

[43] Mahmoudi, E. V., Sabetnia, E., Torshiz, M. N., Jalali, M., & Tabrizi, G. T. (2011, January). Multi-level fuzzy association rules mining via determining minimum supports and membership functions. In 2011, the Second International Conference on Intelligent Systems, Modelling and Simulation (pp. 55-61), IEEE.

[44] Chen, J. S., Chen, F. G., & Wang, J. Y. (2012, August). Enhance the multi-level fuzzy association rules based on the cumulative probability distribution approach. In 2012, the 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (pp. 89-94). IEEE.

[45] Chang, C. I., Chueh, H. E., & Luo, Y. C. (2012, May). An integrated sequential pattern mining with fuzzy time intervals. In 2012 International Conference on Systems and Informatics (ICSAI2012) (pp. 2294-2298). IEEE.

[46] Watanabe, T., & Fujioka, R. (2012, October). Fuzzy association rules mining algorithm based on equivalence redundancy of items. In 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1960-1965). IEEE.

[47] Hong, T. P., Lan, G. C., Lin, Y. H., & Pan, S. T. (2013). An Effective Gradual Data-Reduction Strategy for Fuzzy Itemset Mining. International Journal of Fuzzy Systems, 15(2).

[48] Hong, T. P., Lin, C. W., & Lin, T. C. (2014). The Mffp-Tree Fuzzy Mining Algorithm To Discover Complete Linguistic Frequent Itemsets. Computational Intelligence, 30(1), 145-166.

[49] Lin, J. C. W., Hong, T. P., & Lin, T. C. (2015). A CMFFP-tree algorithm to mine multiple fuzzy frequent itemsets. Applied Soft Computing, 28, 431-439.

[50] Lin, J. C. W., Hong, T. P., Lin, T. C., & Pan, S. T. (2015). A UBMFFP tree for mining multiple fuzzy frequent itemsets. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 23(06), 861-879.

[51] Lin, J. C. W., Li, T., Fournier-Viger, P., & Hong, T. P. (2015). A fast algorithm for mining fuzzy frequent itemsets. Journal of Intelligent & Fuzzy Systems, 29(6), 2373-2379.

[52] Li, H., Zhang, Y., Hai, M., & Hu, H. (2018). Finding Fuzzy Close Frequent Itemsets from Databases. Procedia computer science, 139, 242-247.

[53] Kar, S., & Kabir, M. M. J. (2019). Comparative analysis of mining fuzzy association rule using genetic algorithm. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-5). IEEE.

[54] Srivastava, D. K., Roychoudhury, B., & Samalia, H. V. (2019). Fuzzy association rule mining for economic development indicators. International Journal of Intelligent Enterprise, 6(1), 3-18.

[55] Wang, L., Ma, Q., & Meng, J. (2019). Incremental fuzzy association rule mining for classification and regression. IEEE Access, 7, 121095-121110.

[56] Dhanaseelan, F. R., & Sutha, M. J. (2021). Detection of breast cancer based on fuzzy frequent itemsets mining. Irbm, 42(3), 198-206.

[57] Mendel, J. M., & John, R. B. (2002). Type-2 fuzzy sets made simple. IEEE Transactions on fuzzy systems, 10(2), 117-127.

[58] Castillo, O., Melin, P., Castillo, O., & Melin, P. (2008). 1 Introduction to Type-2 Fuzzy Logic. Type-2 Fuzzy Logic: Theory and Applications, 1-4.

[59] Hagras, H. (2008). Type-2 fuzzy logic controllers: a way forward for fuzzy systems in real-world environments. In Computational Intelligence: Research Frontiers: IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008, Plenary/Invited Lectures (pp. 181-200). Springer Berlin Heidelberg.

[60] Chen, C. H., Hong, T. P., & Li, Y. (2015). Fuzzy association rule mining with type-2 membership functions. In Intelligent Information and Database Systems: 7th Asian Conference, ACIIDS 2015, Bali, Indonesia, March 23-25, 2015, Proceedings, Part II 7 (pp. 128-134). Springer International Publishing.

[61] Lin, J. C. W., Lv, X., Fournier-Viger, P., Wu, T. Y., & Hong, T. P. (2016). Efficient mining of fuzzy frequent itemsets with type-2 membership functions. In Intelligent Information and Database Systems: 8th Asian Conference, ACIIDS 2016, Da Nang, Vietnam, March 14–16, 2016, Proceedings, Part II 8 (pp. 191-200). Springer Berlin Heidelberg.

[62] Bayardo, R. (2014). Frequent itemset mining dataset repository. UCI datasets and PUMSB.

[63] Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. The SPMF Open-Source Data Mining Library Version 2.

# List of Publications

1. Patel, Mahendra Narottamdas, Sanjay M. Shah, and Suresh B. Patel. "An Adjacency matrix-based Multiple Fuzzy Frequent Itemsets mining (AMFFI) technique." International Journal of Intelligent Systems and Applications in Engineering 10, no. 1 (2022): pp. 69–74.
   **(SCOPUS Approved, ISSN: 2147–6799)**

2. Patel, Mahendra Narottamdas, Sanjay M. Shah, and Suresh B. Patel. "An Efficient (MFFPA-2) Multiple Fuzzy Frequent Patterns Mining with Adjacency Matrix and Type-2 Member Function." In *International Conference on Advances in Computing and Data Sciences*, pp. 502-515. Cham: Springer Nature Switzerland, 2023. **(SCOPUS Approved, ISSN: 1865-0929)**