Efficient Parallel Algorithm for Extracting Fuzzy-Crisp Formal Concepts

Ebtesam Shemis[1], Arabi Keshk[2], Ammar Mohammed[3], Gamal Elhady[4]
Department of Computer Science-Faculty of Computers and Information, Menoufia University, Shebin El-Kom City, Egypt[1, 2, 4]
Department of Computer Science-Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt[3]

***Abstract*—Fuzzy Formal Concept Analysis (FFCA) is a robust mathematical tool for analyzing data, particularly where uncertainty or fuzziness is inherent. FFCA is utilized across various domains, including data mining, information retrieval, and knowledge representation. However, fuzzy concepts extraction is a crucial yet computationally intensive task. This paper addresses the challenge of time efficiency in extracting single-sided fuzzy concepts from large datasets. A parallel algorithm is proposed to reduce computational time and optimize resource utilization, thus enabling the scalable analysis of expanding datasets. By computing fuzzy concepts across multiple threads in parallel, each thread processes an attribute independently to extract fuzzy concepts, which are then merged in the final step. The proposed algorithm extracts fuzzy-crisp concepts, which are more concise than other types of fuzzy concepts. Experiments were conducted to evaluate the performance of the proposed parallel algorithm against existing sequential methods. Experimental results demonstrate significant gains in computational efficiency, with the algorithm achieving an average time reduction of 68% compared to the attribute-based algorithm and up to 83%-time reduction compared to the fuzzy CbO algorithm across various types of datasets, including binary, quantitative, and fuzzy.**

Keywords—Fuzzy Formal Concept Analysis; single-sided fuzzy concept; fuzzy-crisp concepts; parallel algorithm; fuzzy concepts extraction; knowledge representation

I. Introduction

Formal Concept Analysis (FCA) is a mathematical framework that involves the systematic study of formal concepts and their interrelationships, providing a structured approach to analyzing complex data sets [1] [2]. Therefore, FCA has found diverse applications across various domains, including linguistics [3], information retrieval [4], bioinformatics [5], Text classification [36], and beyond [6][34]. Traditional FCA algorithms can process only binary datasets [15], namely formal context, to extract formal concepts, constituting logical clusters of related objects and attributes [7]. This framework is based on the principles of crisp set theory, which categorizes objects as either fully belonging to a set or completely outside of it, emphasizing clear boundaries in the conceptual space [35]. Besides, classical FCA algorithms use the crisp scaling method to handle quantitative or qualitative data (many-valued contexts MVC) [8]. As a result, the crisp boundary problem in crisp scaling introduces challenges in accurately defining the boundaries between sub-attributes, affecting the precision of the analysis [9].

Fuzzy Formal Concept Analysis (FFCA) effectively addresses the limitations of handling many-valued and fuzzy contexts by managing ambiguity and uncertainty similarly to human cognition[10]. It represents data on a spectrum rather than in binary terms, allowing for nuanced and accurate analysis of large datasets. FFCA captures the complexity of real-world data, leading to more meaningful insights and clear, interpretable results. Its flexibility allows adaptation to various data mining tasks, making it ideal for applications like customer feedback analysis, where understanding human sentiment is essential. Additionally, FFCA's ability to represent imprecise data aligns with subjective user views, making it suitable for everyday data handling [11] [28]. For example, in the education domain, a university using FFCA to analyze student course evaluations can handle nuanced feedback such as "somewhat satisfied", "mostly satisfied," and "highly satisfied", rather than just "satisfactory" or "unsatisfactory." This deeper understanding of student opinions allows for more targeted improvements in teaching methods and course content, enhancing the overall educational experience.

Extracting the entire set of fuzzy concepts from large datasets is a time-consuming task, recognized as a #P-complete problem [12]. However, if the relationship between object and attribute sets is sparse, even in large datasets, the complexity of this process can be reduced [11]. The generation time of classical formal concepts is often notably shorter than the extraction time of fuzzy concepts. This discrepancy arises due to the inherent complexity involved in computing fuzzy concepts, which typically requires additional computational resources and processing steps [13]. FCA primarily deals with crisp binary relations, leading to relatively faster computations, whereas FFCA involves handling fuzzy relations, which require more complex calculations to derive fuzzy concepts. Therefore, while FFCA handles different data types effectively, it requires extensive time than the traditional FCA that hinder its applicability across various domains. Addressing this research gap, the proposed algorithm significantly enhances the efficiency of generating fuzzy concepts by leveraging parallel processing techniques.

The proposed algorithm differs from In-Close4b [32], bit-close4 [30], and FPCbO [31] in its ability to effectively process a variety of data types. While these algorithms are limited to handling binary data, real-world applications typically involve heterogeneous datasets. In contrast, the proposed algorithm leverages fuzzy set theory, enabling it to process diverse data types, including crisp, vague, and quantitative.

FuzzyInClose4b algorithm [24] generates concepts where both the extent and intent are fuzzy, resulting in an excessive number of fuzzy concepts. This abundance hinders its practical application in data-intensive domains such as association rule extraction, semi-automatic ontology construction, and information retrieval [16]. Conversely, the proposed algorithm generates fuzzy concepts with fuzzy extents and crisp intents, reducing the number of extracted fuzzy concepts. Additionally, the proposed algorithm leverages parallel processing to significantly reduce execution time. These enhancements improve the algorithm's applicability and efficiency.

Through this research, a time-efficient algorithm is proposed for extracting fuzzy concepts from large datasets utilizing parallel processing. The main contributions of this work are summarized as follows:

1) The proposed algorithm utilizes fuzzy logic in extracting fuzzy-crisp concepts from various data types, unlike existing crisp algorithms such as In-Close4b [32], bit-close4 [30], and FPCbO [31], which only process binary data.

2) The proposed algorithm generates more concise fuzzy concepts compared to FuzzyInClose4b [24], facilitating the practical application of FFCA in real-world scenarios.

3) The proposed algorithm reduces the computational time of extracting fuzzy concepts by leveraging multithreading in processing multiple attributes simultaneously to extract fuzzy concepts in parallel, thereby enhancing efficiency and performance. Therefore, it utilizes the available resources for faster computation of fuzzy concepts.

The remainder of this paper is organized as follows: Section II offers a comprehensive review of fundamental notions and definitions concerning formal concept analysis (FCA) and its fuzzy variant. Subsequently, Section 0 takes a closer look at the related algorithms employed in the derivation of formal concepts. Next, Section IV presents the proposed parallel algorithm for generating fuzzy formal concepts, clarifying its operation. Section V introduces a practical case that utilizes the proposed algorithm to see how well it performs in practical in comparison to other algorithms. Section VI evaluates the proposed algorithm across different types of data, presenting a comparative study with other related algorithms. This section highlights the superior performance of the proposed algorithm through detailed analysis. Lastly, Section TABLE IX. provides a summary of the main findings, highlights the key contributions, and discusses avenues for future research and development.

II. Preliminaries

This section provides an overview of the basic definitions and notions related to classical Formal Concept Analysis (FCA) and its fuzzy variant FFCA. Further definitions and foundations can be found in the references [14] for FCA and [17], [18] for FFCA [17].

A. Classical Formal Concept Analysis

Formal concept analysis primarily aims to identify groupings of entities and their associated attributes within a given dataset. Typically, FCA operates on binary formal contexts as its input, wherein a crisp relation I outlines the associations between entities G and their attributes M. Accordingly, a formal context can be concisely characterized as a tabular representation comprising rows denoting entities and columns representing attributes (or vice versa).

Definition 1: A formal context is delineated as a triple $\mathbb{K} = (\mathrm{G}, \mathrm{M}, \mathrm{I} \in \{0, 1\})$, where G signifies the set of objects, M denotes the set of attributes, and I denotes a crisp relation among objects and attributes, where $\mathrm{I} \subseteq \mathrm{G} \times \mathrm{M}$. If an object $\mathrm{g} \in \mathrm{G}$ and an attribute $\mathrm{m} \in \mathrm{M}$ are related in I, it is expressed as the pair $(\mathrm{g}, \mathrm{m}) \in \mathrm{I}$ or $(\mathrm{g}\,\mathrm{I}\,\mathrm{m})$.

Definition 2: A formal concept is denoted as a pair (A, B) of the formal context $\mathbb{K} = (\mathrm{G}, \mathrm{M}, \mathrm{I})$ iff $\mathrm{A} \subseteq \mathrm{G}$, $\mathrm{B} \subseteq \mathrm{M}$, $\mathrm{A}\uparrow = \mathrm{B}$, and $\mathrm{B}\downarrow = \mathrm{A}$, where A and B the concept extent and intent, respectively. And $\mathrm{A}\uparrow$, $\mathrm{B}\downarrow$ are given by Eq. (1) and Eq. (2):

$$A \uparrow := \{\, m \in M \mid (g, m) \in I \;\; \forall\, g \in A \} \tag{1}$$

Such that $A\uparrow$ represents the set of characteristics in M that are shared by all objects in A.

$$B \downarrow := \{\, g \in G \mid (g, m) \in I \; \forall\, m \in B \} \tag{2}$$

Given that $B\downarrow$ is set of objects having all features in the set B.

Classical FCA is typically proficient in processing crisp binary contexts, a scenario that may not align with the inherent complexity of real-world datasets. One challenge encountered pertains to specifying sharp boundaries within scaled attribute intervals, thus posing limitations in effectively addressing the many valued contexts inherent in many datasets.

Fuzzy Formal Concept Analysis (FFCA), in contrast, offers a versatile framework capable of accommodating various types of data, thereby addressing the limitations associated with classical FCA. Unlike its crisp counterpart, fuzzy FCA embraces the inherent uncertainty and vagueness present in real-world datasets by allowing for graded membership degrees. FFCA can handle different types of data (continuous, discrete, or hybrid) types without needing strict boundaries between attribute intervals. Consequently, FFCA offers a more robust and adaptable approach for analyzing complex datasets characterized by imprecision and ambiguity.

B. Fuzzy Formal Concept Analysis

Fuzzy Formal Concept Analysis (FFCA) is a mathematical framework designed to analyze and extract meaningful patterns from datasets characterized by uncertainty and imprecision. Fuzzy sets [33] in FFCA replace the conventional binary representation of relationships between objects and attributes where a value in the range [0, 1] represents the degree of membership of an object to an attribute.

Definition 3: a fuzzy formal context is a triple $\widehat{\mathbb{K}} = (\mathrm{G}, \mathrm{M}, \hat{\mathrm{I}} \in [0, 1])$, where G is a set of objects, M is a set of attributes, $\hat{\mathrm{I}}$ is a fuzzy relation between objects and attributes each with membership μ that gives each pair (g, m) in G × M a degree of membership. This membership degree shows how strongly objects and attributes are linked.

FFCA extends traditional FCA by allowing for the representation of graded relationships, enabling a more flexible and nuanced analysis of complex datasets. One-sided FFCA involves the fuzzification of either the extent or the intent of a formal concept while maintaining a crisp definition on the other side [10][17].

As depicted in Fig. 1, various viewpoints and formal definitions for fuzzy concepts exist. Fuzzy concepts can be categorized into single-sided fuzzy concepts and full-sided fuzzy concepts. In single-sided fuzzy concepts, only the intent or extent of the fuzzy concept is fuzzy, not both [26]. On the other hand, full-sided fuzzy concepts have both extent and extent represented as fuzzy sets [24]. This paper proposes a parallel algorithm for extracting single-sided fuzzy concepts in which extents are fuzzy and intents are crisp. The rest of this section presents the formal definitions and notions for this type of FFCA.

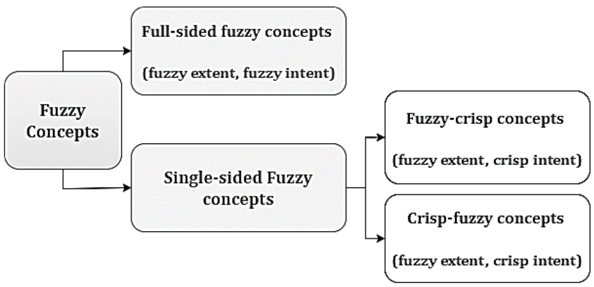

Fig. 1. Fuzzy concepts' taxonomy.

Definition 4: In a fuzzy context $\widehat{\mathbb{K}} = (\mathrm{G}, \mathrm{M}, \hat{\mathrm{I}})$, a fuzzy-crisp concept is represented by a pair $(\mathrm{A_f}, \mathrm{B})$, where $\mathrm{A_f} \subseteq \mathrm{G}$ denotes the fuzzy extent of the concept, $\mathrm{B} \subseteq \mathrm{M}$ represents the crisp intent, and $\mathrm{A_f} \uparrow = \mathrm{B}$ and $\mathrm{B} \downarrow = \mathrm{A}$. The definitions of $\mathrm{A_f} \uparrow$ and $\mathrm{B} \downarrow$ are provided by Eq. (3) and Eq. (4) [17]:

$$A_f \uparrow = \{ b \in B \mid \forall a \in A: \mu_I (a, b) \geq \mu_A (a)\} \tag{3}$$

$$B \downarrow := \{ a \mid \mu_A (a) = min_{\{b \in B\}} (\mu_I (a, b))\} \tag{4}$$

In this context, $A_f \uparrow$, denoted as $A_f{}'$, delineates the crisp intent of the object set A_f, while $\mathrm{B} \downarrow$, denoted as B', delineates the fuzzy extent of the attribute set B. A confidence threshold α is utilized where $\alpha < 1$. The user can adjust the threshold α according to their requirements for generating fuzzy concepts from a provided fuzzy context [20]. This threshold aids in filtering out fuzzy relations that fall outside the interval $[\alpha, 1]$ from the given fuzzy formal context, thereby facilitating knowledge discovery and representation [19] [21] [28]. Alternatively, the threshold interval is denoted as α-cut of the fuzzy context.

Example 1: Table I shows a fuzzy context example that refers to a part of the employee's dataset. According to Def. 3, the attribute set M is $\{A, B, C, D, E, F\}$, and each attribute corresponds to the properties: A: low salary, B: moderate salary, C: high salary, D: young, E: youth, and F: old. Moreover, the object set $G = \{O_0, O_1, O_2, O_3, O_4\}$ represents five different employee instances. Applying an α-cut of 0.4 to the fuzzy context in Table I produces the fuzzy context in Table II. Eliminated relationships are highlighted in gray background in Table II. Applying the threshold clearly produces a sparser context and can effectively eliminate unwanted relationships.

TABLE I. EXAMPLE FUZZY CONTEXT

Obj	A	B	C	D	E	F
O_0	1	0	0	1	0	0
O_1	0.22	0.78	0	1	0	0
O_2	0	0.33	0.67	0.73	0.27	0
O_3	0	0	1	0	0.36	0.64
O_4	0	0	1	0	0	1

TABLE II. FUZZY CONTEXT WITH α-$cut = 0.4$

Obj	A	B	C	D	E	F
O_0	1	0	0	1	0	0
O_1	0	0.78	0	1	0	0
O_2	0	0	0.67	0.73	0	0
O_3	0	0	1	0	0	0.64
O_4	0	0	1	0	0	1

For more clarity, Table III illustrates the fuzzy concepts extracted from the fuzzy context in Table II after applying the threshold $\alpha = 0.4$. As shown in Table III, there are ten fuzzy concepts, each consisting of a fuzzy extent and a crisp intent.

TABLE III. FUZZY CONCEPTS GENERATED FROM THE FUZZY CONTEXT IN TABLE II

#	Fuzzy Extent	Crisp Intent
C_1	$\{O_0: 1, O_1: 1, O_2: 1, O_3: 1, O_4: 1\}$	{}
C_2	$\{O_0: 1\}$	{A, D}
C_3	{}	{A, B, C, D, E, F}
C_4	$\{O_1: 0.78, O_2: 0.33\}$	{B, D}
C_5	$\{O_2: 0.33\}$	{B, C, D}
C_6	$\{O_2: 0.67, O_3: 1, O_4: 1\}$	{C}
C_7	$\{O_2: 0.67\}$	{C, D}
C_8	$\{O_3: 0.36\}$	{C, E, F}
C_9	$\{O_3: 0.64, O_4: 1\}$	{C, F}
C_{10}	$\{O_0: 1, O_1: 1, O_2: 0.73\}$	{D}

A fuzzy extent consists of a set of objects and their corresponding memberships in the form of $\{O_i: \mu_{o_i}\}$, such that O_i refers to the object number i and μ_{o_i} is the membership of O_i to the corresponding crisp intent. For instance, fuzzy concept (4) is $(\{O_1: 0.78, O_2: 0.33\}, \{B, D\})$ where $(\{O_1: 0.78, O_2: 0.33\}$ is the fuzzy extent consisting of objects O_1 and O_2 with membership degrees of 0.78 and 0.33, respectively. These membership degrees determine to what extent the objects possess the attributes in the crisp intent $\{B, D\}$.

III. RELATED WORKS

Extracting formal concepts from formal contexts has gained a significant interest in classical algorithms that handle binary datasets using crisp set theory. In-Close4b [32], bit-close4 [30], and FPCbO [31] algorithms identify crisp concepts through depth-first search and pruning techniques. With a mix of arrays and bitsets, In-Close4b balances memory usage and speed, making it suitable for large datasets. Bit-Close4 enhances performance for sparse data by optimizing memory efficiency with bitset data structures. FPCbO [31] has improved the classical CbO algorithm [23] by reducing computational redundancy and employing a canonical direct basis representation to generate only canonical concepts. Regarding optimizations, these algorithms differ in the following ways: In-Close4b balances resources; Bit-Close4 emphasizes memory efficiency; and FPCbO focuses on redundancy reduction.

Compared to the proposed algorithm, these crisp algorithms are less adaptable to diverse data types and cannot handle fuzzy or imprecise data, limiting their applicability in real-world scenarios where data uncertainty is common. While In-Close4b balances memory usage and speed, it struggles with extremely large or dense datasets, and its complexity makes it harder to implement. Bit-Close4's reliance on bitsets enhances performance for sparse matrices but can reduce flexibility and efficiency in dense datasets. FPCbO's strict canonicality minimizes redundancy but introduces significant computational overhead and increased memory consumption, making it less efficient for complex datasets. Overall, these limitations highlight the need for more versatile and scalable algorithms, offered by fuzzy concept extraction algorithms.

Fuzzy FCA algorithms gained less popularity due to its extensive time requirement despite its capability to process diverse data types, including fuzzy and imprecise data. This section reviews current fuzzy algorithms, identifying key limitations that the proposed algorithm aims to address.

Fuzzy CbO [22] presents an algorithm to extract fuzzy-crisp concepts. Therefore, fuzzy CbO can process fuzzy and vague data effectively. In addition, this recursive algorithm exploits canonicity tests to discover new fuzzy concepts, demonstrating superior efficacy in fuzzy concept generation when compared to the fuzzy NextClosure algorithm[25]. Despite its effectiveness in processing fuzzy data, the fuzzy CbO algorithm faces limitations such as significant computational overhead from canonicity tests, high memory consumption due to its recursive nature, and potential challenges in handling large and diverse datasets. These limitations are efficiently handled by the proposed parallel algorithm.

The attribute-based algorithm [27] excels with symmetrically correlated attributes, outperforming fuzzy CbO in execution time under these conditions. However, in other scenarios, both algorithms demonstrate similar performance, limiting the attribute-based algorithm's effectiveness in handling diverse datasets and asymmetric correlations. On the contrary, the proposed algorithm leverages parallel processing techniques to handle multiple attributes at the same time, which results in efficient processing of fuzzy concepts. Unlike sequential algorithms like Fuzzy CbO and attribute-based algorithm, the proposed algorithm excels in handling large, dense datasets and produces compact fuzzy-crisp concepts efficiently.

FuzzyInClose4b algorithm[24] utilizes incremental closure and matrix searching techniques to extract full-sided fuzzy concepts. It computes each closure incrementally, only once per concept, to prevent repeated closure calculations. Compared to the proposed algorithm, FuzzyInClose4b[24] generate more fuzzy concepts. Therefore, it requires plenty of time and storage space. But it may be more suitable when the intent's membership values are essential. But for Ontology construction applications and association rule mining, it can be sufficient to use fuzzy-crisp concepts algorithms like the proposed algorithm.

To the best of knowledge, all existing algorithms for generating fuzzy-crisp formal concepts operate sequentially, failing to leverage parallel computation. This limitation significantly impacts execution time, particularly with large datasets. This work addresses this gap by proposing a parallel algorithm to accelerate the generation of fuzzy concepts.

IV. Proposed Parallel Algorithm for Computing Fuzzy-Crisp Concepts

This section introduces the proposed parallel fuzzy concept (PFC) algorithm to generate fuzzy-crisp concepts from a fuzzy formal context in parallel, leveraging multi-threading to improve efficiency. The algorithm accesses the fuzzy context and eventually returns fuzzy concepts as pairs of fuzzy extents and crisp intents.

Algorithm: PFC( )

Input: A fuzzy formal context $\widehat{\mathbb{K}} = (\mathrm{G}, \mathrm{M}, \hat{\mathrm{I}} \in [0,1])$
Output: The set of FC of all fuzzy concepts of $\widehat{\mathbb{K}}$
Begin
1 Initialize FC ← $\{(G\uparrow, G), (M\downarrow, M)\}$
2 With ThreadPoolExecutor() as executor:
3 For each i from 0 to $|M|-1$ do
4 executor.map(ProcessAttribute, i)
5 End for
6 Return FC
Procedure: ProcessAttribute ($attr_i$)
1 F_extent ← $attr_i\downarrow$
2 intent ← F_extent ↑
3 If intent ⊈ FC.intents then:
4 FC← $FC \cup$ (F_extent , intent)
5 With threading.lock():
6 For c in FC \ {(F_extent, intent) ∪ (G, G↓)∪ $(M\downarrow, M)$}
7 Inters = Fextent ∧ c.extent
8 If Inters ↓ ⊈ FC.intents then:
9 $FC \leftarrow FC \cup$ (Inters, Inters ↓)
10 End If
11 End For
12 End If
End Procedure

The Parallel Fuzzy Concept (PFC) algorithm begins by initializing the global fuzzy concepts set with the bottom and top concepts (line 1). The top concept is a set of all objects G and their shared intent $G\uparrow$ evaluated using Eq. (3). The bottom concept is a set of whole intent M and all objects that assess M, given by $M\downarrow$ provided using Eq. (4), and represents the fuzzy extent in the form $ext{:}\,\mu_{ext}$. Subsequently, a thread pool is employed to process attributes in parallel (Lines 2–5), where each thread is responsible for computing all fuzzy concepts discoverable using a particular attribute. Therefore, the algorithm ensures efficient computation of fuzzy concepts using multiple threads.

To identify all new fuzzy concepts given an attribute, the ProcessAttribute() procedure is invoked. First, lines $1-2$ calculate the fuzzy extent using Eq. (4) and the crisp intent using Eq. (5). Next, line 3 verifies that the concept intent is new and not present in the FC set, thereby storing the new fuzzy concept in FC (line 4). In line 5, a locking mechanism ensures thread safety during updates. Lines 6–9 then uncover all other fuzzy concepts that arise from intersections between the new fuzzy extent (F_extent) and the existing concepts' extents. Therefore, line 6 presents a loop that iterates over all existing concepts except all fuzzy concepts that cannot produce new concepts via the fuzzy intersection (i.e., the newly generated concept {F_extent, intent}, the top concept $(G, G\uparrow)$, and the bottom concept $(M\downarrow, M)$. In line 7, Zadah's intersection, as stated in definition (5), evaluates the fuzzy intersection between the extent of the new concept and the extent of the existing concept. If the intent of an intersection is not in FC, it adds the new concept to FC (lines 8 and 9).

Definition 5: The intersection of two fuzzy sets A and B with membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively, results in a fuzzy set C, denoted as $C = A \cap B$. The membership function of C is defined as $\mu_C(x) = \mathrm{Min}[\mu_A(x), \mu_B(x)]$ for all $x \in X$ in the universe of discourse X. This can also be represented as $C = A \wedge B$.

The proposed algorithm significantly enhances efficiency by leveraging parallel processing technique (multi-threading), making it well-suited for large datasets. The combination of parallel attribute processing and intersection checking ensures the identification of all fuzzy concepts within the fuzzy context.

Unlike existing algorithms for extracting full-sided fuzzy concepts, such as FuzzyInClose4b [24], the proposed algorithm generates fuzzy-crisp concepts that are more compact and suitable for information retrieval, ontology construction and association rule mining. Additionally, the parallel nature of the proposed algorithm makes it more efficient in extracting fuzzy-crisp concepts from large dense datasets, a feature that is not present in existing sequential fuzzy-crisp concept generation algorithms like Fuzzy CbO [22], Fuzzy NextClosure [25], and attribute-based algorithm [27].

V. Case Study and Discussion

This section demonstrates the practical application of the proposed PFC algorithm in analyzing a fuzzy keyword-document context. This context involves eight keywords (k_1 to k_8) and four documents (d_1 to d_4), where fuzzy values represent the strength of relationships between keywords and documents. Keywords are treated as fuzzy extents, and documents as crisp intents, quantified by fuzzy membership values ranging from 0 to 1, as shown in Table IV.

Using the proposed PFC algorithm, this fuzzy context is systematically analyzed to extract fuzzy-crisp concepts, detailed in Table V. These concepts are characterized by fuzzy extents, indicating the strength of associations between keywords and documents, and crisp intents, describing documents based on keywords.

TABLE IV. Keyword-Document Fuzzy Context

	d_1	d_2	d_3	d_4
k_1	0	1	1	0
k_2	0.2	0.6	1	0.2
k_3	1	0	0	0
k_4	0.3	0.6	0.7	0.3
k_5	1	0	0	1
k_6	0	1	0	0
k_7	0.1	0.5	0.5	0.1
k_8	0.5	0.3	0	0.5

TABLE V. Fuzzy-Crisp Fuzzy Concepts Extracted by the Proposed Algorithm

#	Fuzzy Extents (keywords)	Crisp Intents (documents)
$C1$	$\{k_0: 1, k_1: 1, k_2: 1, k_3: 1, k_4: 1, k_5: 1, k_6: 1, k_7: 1\}$	$\{\ \}$
$C2$	$\{k_1: 0.2, k_3: 0.3, k_6: 0.1\}$	$\{d_1, d_2, d_3, d_4\}$
$C3$	$\{k_1: 0.2, k_2: 1, k_3: 0.3, k_4: 1, k_6: 0.1, k_7: 0.5\}$	$\{d_1\}$
$C4$	$\{k_0: 1, k_1: 0.6, k_3: 0.6, k_5: 1.0, k_6: 0.5, k_7: 0.3\}$	$\{d_2\}$
$C5$	$\{k_1: 0.2, k_3: 0.3, k_6: 0.1, k_7: 0.3\}$	$\{d_1, d_2, d_4\}$
$C6$	$\{k_0: 1, k_1: 1, k_3: 0.7, k_6: 0.5\}$	$\{d_3\}$
$C7$	$\{k_0: 1, k_1: 0.6, k_3: 0.6, k_6: 0.5\}$	$\{d_2, d_3\}$
$C8$	$\{k_1: 0.2, k_3: 0.3, k_4: 1, k_6: 0.1, k_7: 0.5\}$	$\{d_1, d_4\}$

Fig. 2 illustrates the steps of the PFC algorithm. Initially, the top and bottom fuzzy concepts (C1 and C2) are added to the fuzzy concepts set (line 1 of the algorithm). According to line 2, four threads are utilized to process different attributes (d1, d2, d3, d4) in parallel. Each thread independently processes its assigned attribute by calculating the fuzzy extent and corresponding crisp intent and checks if the newly formed concept's intent is not in the set of fuzzy concepts (FC). If it's unique, it adds a new concept to FC. Additionally, each thread performs fuzzy intersections of extents with existing concepts and ensures that new intersection concepts, if unique, are also added to FC. For example, processing attribute d_1 results in a concept with the fuzzy extent $\{k_1: 0.2,\ k_2: 1,\ k_3: 0.3, k_4: 1,\ k_6: 0.1,\ k_7: 0.5\}$ and the crisp intent $\{d_1\}$.

This process is repeated for all attributes, ensuring systematic extraction of all fuzzy concepts. This tracing highlights the algorithm's parallel behavior, where multiple threads execute concurrently to process different attributes, significantly speeding up the concept generation process by leveraging parallel computation. By analyzing these fuzzy concepts, the varying strengths of relationships between keywords and documents can easily be demonstrated.

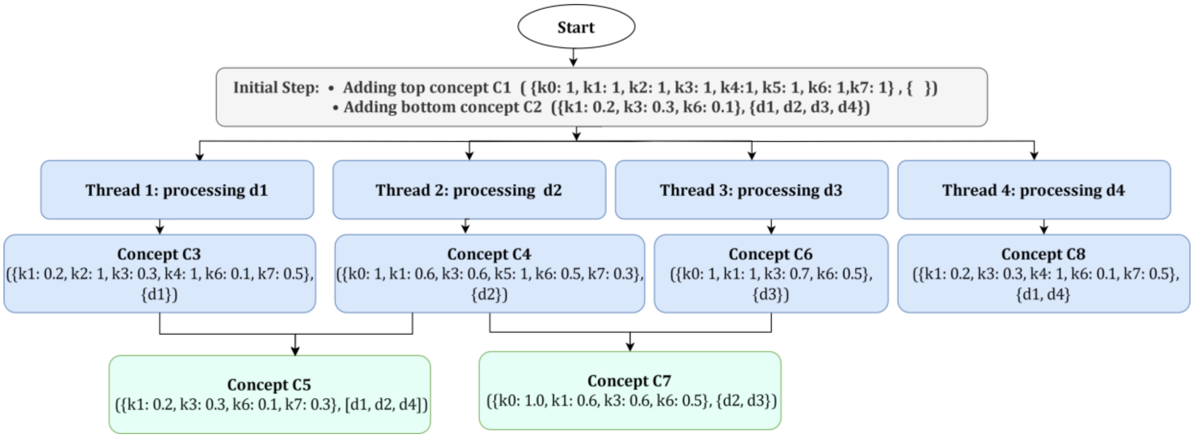

Fig. 2. Steps by the proposed PFC algorithm for extracting fuzzy concepts from fuzzy context in Table IV.

This analysis helps identify documents with strong, weak, or similar associations to multiple keywords. For instance, fuzzy concept C8 highlight that document d_1 and d_4 have a strong association with keyword k_4 and moderate to weak associations with other keywords. These keywords serve as a linking set for these two documents. Information from fuzzy-crisp concepts is valuable for tasks like information retrieval, as it improves search accuracy by highlighting documents strongly associated with the user query. The FuzzyInClose4b [24] algorithm generated 43 fuzzy concepts from Table IV in which both extent and intent are fuzzy. Example fuzzy concepts are: $(\{k_2, k_4, k_7, k_8{:}\ 0.3\},\ \{d_1{:}\,0.1, d_2{:}\,0.5, d_4{:}\,0.1\})$ and $(\{k_2, k_4, k_7{:}\ 0.1, k_8{:}\ 0.3\},\ \{d_1{:}\,0.2,\ d_2{:}\,0.6,\ d_4{:}\,0.2\})$ which are very similar concepts.

In this case study, fuzzy-crisp concepts extracted by the proposed PFC algorithm, which use fuzzy extents for keyword relevance and crisp intents for document associations, provide an efficient and nuanced approach. This method captures keyword importance precisely while maintaining clear document associations, leading to effective and context-aware search results. For example, in a search for "renewable energy sources," relevant keywords can be weighted, and documents are crisply linked, enhancing search efficiency. In contrast, full sided fuzzy concepts extracted by FuzzyInClose4b [24] offer more detailed relationships by representing both keywords and documents with varying degrees, but they introduce higher computational complexity and interpretation challenges with the existence of very similar fuzzy concepts. Thus, the fuzzy-crisp concepts extracted by the proposed PFC algorithm balance complexity and precision, delivering relevant and personalized search outcomes efficiently.

VI. Results and Discussion

This section shows the efficiency and scalability of the proposed PFC algorithm through experiments on synthetic and benchmark datasets. The aim of these experiments is to show that the proposed PFC algorithm significantly reduces computation time, making FFCA feasible for large-scale data analysis. Table VI provides an overview of the datasets utilized in the experiments. The Iris and red-wine datasets are benchmarks from the UCI Repository [29]. To further assess the robustness of the proposed algorithm, we synthesized three variations of a fuzzy dataset with different densities (20%, 30% and 40%) and a size of (20,000 × 15), so the performance of the proposed algorithm under various conditions can be assessed.

TABLE VI. Datasets Used in the Experiments

Dataset	$\|G\|$	$\|M\|$	Density	Description
Fuzzy Red Wine	1600	36	66.2%	Multivariate (fuzzified)
Fuzzy Iris	150	15	58%	Multivariate (fuzzified)
Synthetic Fuzzy Dataset	20,000	15	20%, 30%, 40%	Synthetic Fuzzy
Car	1,728	25	28%	Crisp

To maintain consistency with the fuzzy setting, all quantitative attributes are fuzzified using three linguistic labels per attribute: low, moderate, and high. Trapezoidal membership functions are employed to represent the low and high linguistic labels, while a triangular membership function is used for the moderate label. The proposed PFC algorithm is implemented in Python, and the experiments are carried out on a Windows machine equipped with an Intel Core i7 processor running at 2.60 GHz and 32 GB of RAM.

Before exploring experiments, Table VII presents feature comparisons between the proposed PFC algorithm and the related algorithms: In-Close4b [32], bit-close4 [30], FPCbO [31], FuzzyInClose4b [24], fuzzy CbO [22], attribute-based algorithm [27].

TABLE VII. Features Comparison

Algorithms	Data Types	Parallel	Concept Extent	Concept Intent
In-Close4b [32]		x		
bit-close4 [30]	Binary	✓	Crisp	Crisp
FPCbO [31]		✓		
FuzzyInClose4b[24]		x	Fuzzy	Fuzzy
fuzzy CbO [22]	Fuzzy	x		
attribute-based [27]		x	Fuzzy	Crisp
PFC algorithm	Fuzzy	✓		

The algorithms In-Close4b [32], bit-close4 [30], and FPCbO [31] can only extract formal concepts from binary data, such as Car dataset. They can't process other types of data, such as multivariate and fuzzy datasets. In contrast, the proposed PFC algorithm can process these data by utilizing fuzzy set theory in computing fuzzy concepts.

The FuzzyInClose4b algorithm [24] extracts full-sided fuzzy concepts, which are more numerous than fuzzy-crisp concepts, extracted by the proposed PFC algorithm. Table VIII compares the number of fuzzy concepts extracted by the FuzzyInClose4b algorithm with those extracted by the proposed PFC algorithm for the fuzzy Iris dataset considering different α-cuts. Due to the larger count of concepts, the FuzzyInClose4b algorithm is computationally intensive and demands significant storage resources.

TABLE VIII. COMPARISON OF FUZZY CONCEPTS BETWEEN FUZZY IN CLOSE4B AND PROPOSED PFC ALGORITHMS ON FUZZY IRIS DATASET

$\alpha - cut$	full-sided fuzzy concepts	Fuzzy-crisp concepts
0.8	844	73
0.7	2,738	87
0.6	8,438	110
0.5	16,843	160
0.4	46,514	252

All of fuzzy CbO [22], attribute-based algorithm [27], and the PFC algorithm extract fuzzy-crisp concepts. However, both fuzzy CbO and attribute-based algorithms operate sequentially and are computationally intensive. In contrast, the proposed PFC algorithm leverages parallel processing of attributes, significantly enhancing its efficiency. Experiments are conducted to compare the computational time (in seconds) required by fuzzy CbO, the attribute-based algorithm, and the PFC algorithm at different α-cut values applied to the dataset.

Fig. 3–6 demonstrate the persistent effectiveness of the suggested parallel algorithm in comparison to other algorithms. For all algorithms, the computation times decrease as the α-cut threshold increases. Aside from that, the proposed PFC algorithm consistently outperforms the other algorithms, achieving the shortest calculation times across all thresholds.

Fig. 3 demonstrates a performance comparison over the fuzzy Iris dataset, indicating that the suggested PFC algorithm consistently takes the least amount of time to compute all fuzzy concepts for all α-cuts applied to the dataset. Although the sequential attribute-based approach and the PFC algorithm behave similarly on this dataset, the difference in performance between them becomes more apparent when the α-cut decreases. This indicates that the suggested parallel algorithm is especially suitable for datasets with a high density.

Fig. 4, 5, and 6 display the efficiency for the synthetic fuzzy dataset with densities of 20%, 30%, and 40%, respectively. The proposed PFC algorithm consistently outperforms the fuzzy CbO and attribute-based algorithms on synthetic fuzzy datasets with different densities. It also has a stable and efficient execution time of 5 to 40 seconds, which doesn't change depending on the α-cut thresholds. On the other hand, the density of the dataset significantly influences the Fuzzy CbO and attribute-based algorithms. Regardless, they both show a decrease in execution time with increasing α-cut values but remain inefficient. The proposed parallel algorithm is always efficient, even when datasets are very dense.

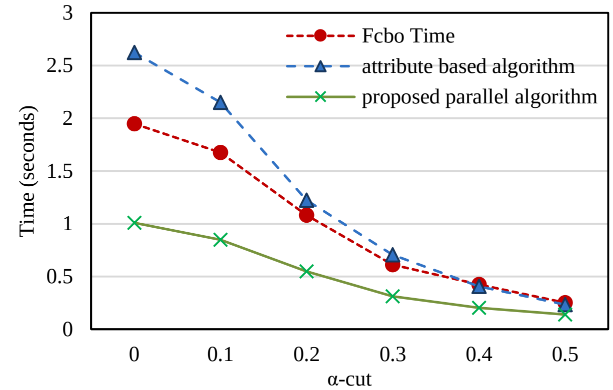

Fig. 3. Performance comparison of algorithms on fuzzy Iris dataset.

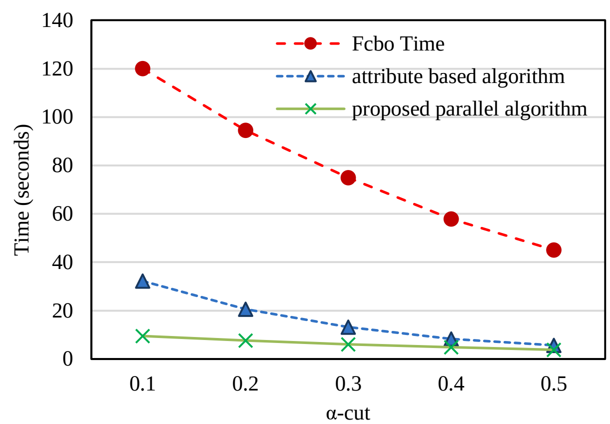

Fig. 4. Performance of algorithms on the synthetic fuzzy dataset with 20% density.

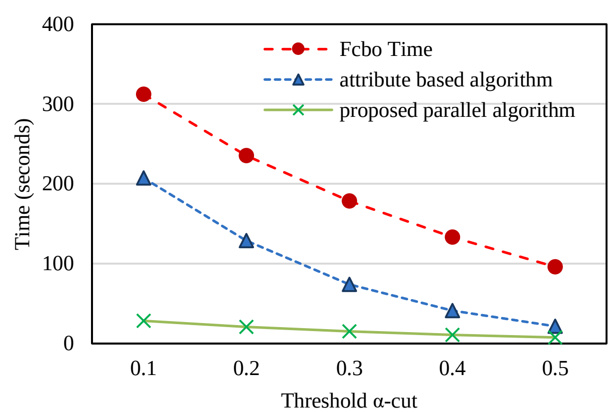

Fig. 5. Performance of algorithms on the synthetic fuzzy dataset with 30% density.

TABLE IX. EXPERIMENTS OVER DIVERSE DATASETS

Expr #	dataset	α-cut	Concepts#	Fuzzy CbO algorithm	Attribute based algorithm	Proposed parallel algorithm	Time reduction w. r. t. attribute-based algorithm	Time reduction w. r. t. Fuzzy CbO algorithm
1	Fuzzy Iris	0	730	2.15	1.38	**1.00**	27.89%	53.63%
2		0.1	693	1.63	1.13	**0.88**	22.07%	45.92%
3		0.2	589	1.11	0.68	**0.55**	19.40%	50.57%
4		0.3	376	0.65	0.40	**0.33**	16.46%	48.74%
5		0.4	252	0.42	0.24	**0.20**	18.12%	52.08%
6		0.5	160	0.26	0.16	**0.14**	12.89%	46.51%
Average reduction							**19%**	**50%**
7	synthetic fuzzy dataset with density 20%	0	7,124	187.39	32.45	**10.13**	68.79%	94.60%
8		0.1	7,124	120.14	32.18	**9.55**	70.32%	92.05%
9		0.2	5,650	94.54	20.58	**7.63**	62.95%	91.93%
10		0.3	4,397	74.93	13.25	**6.10**	53.97%	91.86%
11		0.4	3,292	57.89	8.36	**4.86**	41.86%	91.61%
12		0.5	2,503	45.14	5.63	**3.80**	32.51%	91.59%
Average reduction							**55%**	**92%**
13	synthetic fuzzy dataset with density 30%	0	17,688	431.76	210.47	**30.03**	85.73%	93.05%
14		0.1	17,688	312.39	207.38	**28.43**	86.29%	90.90%
15		0.2	13,622	235.24	128.90	**20.49**	84.10%	91.29%
16		0.3	10,353	178.58	74.03	**14.85**	79.94%	91.68%
17		0.4	7,633	132.99	41.15	**10.64**	74.14%	92.00%
18		0.5	5,630	95.93	21.43	**7.52**	64.90%	92.16%
Average reduction							**79.19%**	**91.85%**
19	synthetic fuzzy dataset with density 40%	0	28,834	734.45	673.38	**86.79**	87%	88%
20		0.1	28,834	829.09	773.33	**106.27**	86%	87%
21		0.2	24,997	679.03	632.23	**69.29**	89%	90%
22		0.3	20,229	402.54	307.51	**34.35**	89%	91%
23		0.4	15,045	255.41	164.50	**22.35**	86%	91%
24		0.5	10,652	174.43	77.96	**14.57**	81%	92%
Average reduction							**86.49%**	**89.92%**
25	Fuzzy Red Wine	0.9	356	3.34	0.49	**0.39**	**20%**	**88%**
26		0.8	2,909	19.82	13.06	**2.90**	**78%**	**85%**
27		0.7	9,822	85.13	250.43	**14.68**	**94%**	**83%**
28		0.6	27,455	454.51	2481.24	**62.15**	**97%**	**86%**
Average reduction							**62.71%**	**72.09%**
29	car	0	12,640	95.92	461.29	**5.59**	**98.79%**	**94.17%**

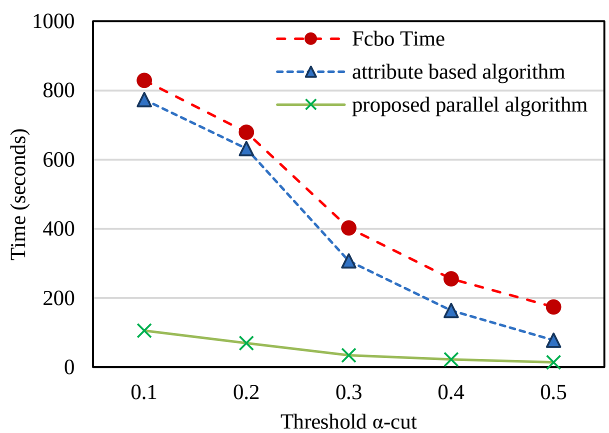

Fig. 6. Performance of algorithms on the synthetic fuzzy dataset with 40% density.

Table IX compares the proposed parallel algorithm's efficiency across diverse datasets, showing significant time reductions compared to the attribute-based and Fuzzy CbO algorithms. For instance, in the synthetic fuzzy dataset with 20% density, the proposed algorithm achieved an average time reduction of 55% compared to the attribute-based algorithm and 92% compared to the Fuzzy CbO algorithm, demonstrating its robustness and superior performance.

VII. CONCLUSION

Fuzzy formal concepts are essential to reveal the fundamental structures and patterns in heterogeneous datasets, thus enabling efficient decision-making in several fields. However, due to the complexity and ambiguity inherent in fuzzy datasets, extracting fuzzy concepts is computationally intensive. This inefficiency can hinder the practical application of fuzzy concepts in the real world. To address this challenge, the proposed novel parallel algorithm optimizes the extraction of fuzzy-crisp concepts from fuzzy datasets, exploiting the resources at disposal. The proposed PFC algorithm utilized multi-threading to process attributes in parallel, then merge extracted fuzzy concepts. Experiments witness that the proposed algorithm reduces computation times and improves scalability, making it more suitable for handling dense and complex data structures. Besides, performance evaluation of the proposed PFC algorithm against other fuzzy algorithms across multiple datasets has demonstrated the algorithm's consistent superiority, achieving the shortest processing times across all α-cuts, with notable efficiency improvements as the α-cut decreases.

Future research should enhance the algorithm's applicability in various computational environments, including big data processing frameworks like Apache Spark, and explore its adaptation for distributed computing environments, allowing workload division and reduced computation times. Besides, a comprehensive evaluation of the algorithm's performance across a wider range of datasets and use cases will provide deeper insights into its strengths and limitations, guiding future enhancements and adaptations. Besides, the algorithm's efficient fuzzy concept extraction could revolutionize applications in data processing and decision-making, including online recommendation systems and dynamic risk assessment.

REFERENCES

[1] R. Wille, “Restructuring lattice theory: an approach based on hierarchies of concepts”, *Ordered sets. Springer*. pp. 445-470, 1982.

[2] B. Breckner, C. Săcărea, and R. R. Zavaczki, "Improving User's Experience in Exploring Knowledge Structures: A Gamifying Approach," *Mathematics*, vol. 10, no. 5, p. 709, 2022.

[3] M. Bogatyrev and D. Orlov, "Application of formal contexts in the analysis of heterogeneous biomedical data," In: *CEUR Workshop Proc.*, vol. 2648, pp. 315-329, 2020.

[4] M. Tropmann-Frick, "Improvement of Searching for Appropriate Textual Information Sources Using Association Rules and FCA," in *Information Modelling and Knowledge Bases XXXIII*, vol. 343, p. 204, 2022.

[5] S. Roscoe et al., "Formal concept analysis applications in bioinformatics," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1-40, 2022.

[6] C. M. Rocco, E. Hernandez-Perdomo, and J. Mun, "Introduction to formal concept analysis and its applications in reliability engineering," *Reliability Engineering & System Safety*, vol. 202, p. 107002, 2020.

[7] L. Zhao, L. X. Lu, and W. Yao, "Generalized three-way formal concept lattices," *Soft Computing*, vol. 27, no. 16, pp. 11219-11226, 2023.

[8] R. Belohlavek, E. Sigmund, and J. Zacpal, "Evaluation of IPAQ questionnaires supported by formal concept analysis," *Information Sciences*, vol. 181, no. 10, pp. 1774-1786, 2011.

[9] K. Sumangali and A. K. Ch, "Concept lattice simplification in formal concept analysis using attribute clustering," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 6, pp. 2327-2343, 2019.

[10] E. Shemis, G. Elhady, A. Mohammed and A. Keshk, "A Fuzzy-Crisp Frequent Concept Lattice Generation Algorithm," 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), IEEE, Cairo, Egypt, 2022, pp. 75-80, doi: 10.1109/MIUCC55081.2022.9781692.

[11] E. Shemis, A. Gadallah, and H. Hefny, "A Data-Sensitive Approach for Fuzzy Concept Extraction," *International Journal of Intelligent Engineering & Systems*, vol. 11, no. 5, 2018.

[12] E. Shemis and A. Mohammed, "A comprehensive review on updating concept lattices and its application in updating association rules," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 2, p. e1401, 2021.

[13] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene, "Fuzzy and rough formal concept analysis: a survey," *International Journal of General Systems*, vol. 43, no. 2, pp. 105-134, 2014.

[14] B. Ganter and R. Wille, "Formal concept analysis: mathematical foundations," *Springer Science & Business Media*, 2012.

[15] R. Wille, "Formal concept analysis as mathematical theory of concepts and concept hierarchies," in *Formal concept analysis: Foundations and applications*, B. Ganter and S. Obiedkov, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1-33.

[16] C. Aswani Kumar, S. Mouliswaran, P. Amriteya, and S. Arun, "Fuzzy formal concept analysis approach for information retrieval," in *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015)*, Springer International Publishing, 2015, pp. 255-271.

[17] A. Majidian, T. Martin, and M. E. Cintra, "Fuzzy formal concept analysis and algorithm," in *Proceedings of the 11th UK Workshop on Computational Intelligence*, Citeseer, September 2011, pp. 61-67.

[18] S. Boffa, "Extracting concepts from fuzzy relational context families," IEEE Transactions on Fuzzy Systems, vol. 31, no. 4, pp. 1202-1213, 2022.

[19] P.K. Singh, C. Aswani Kumar, and J. Li, "Knowledge representation using interval-valued fuzzy formal concept lattice," *Soft Computing*, vol. 20, pp. 1485–1502, 2016, doi: 10.1007/s00500-015-1600-1.

[20] W. Khemili, J. E. Hajlaoui, and M. N. Omri, "Energy aware fuzzy approach for placement and consolidation in cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 161, pp. 130-142, 2022.

[21] A. Izadpanahi, "Relational Approach to the L-Fuzzy Concept Analysis," master's thesis, *Brock Univ*ersity, St. Catharines, ON, Canada, 2023.

[22] T. Martin and A. Majidian, "Finding Fuzzy Concepts for Creative Knowledge Discovery," in *IEEE Transactions on Intelligent Systems*, vol. 28, pp. 93-114, 2013. https://doi.org/10.1002/int.21576

[23] P. Krajca, J. Outrata, and V. Vychodil, "Parallel algorithm for computing fixpoints of Galois connections," *Annals of Mathematics and Artificial Intelligence*, vol. 59, no. 2, pp. 257-272, 2010.

[24] D. López-Rodríguez, Á. Mora, and M. Ojeda-Hernández, "Revisiting Algorithms for Fuzzy Concept Lattices," in *Proceedings of the International Conference on Concept Lattices and Their Applications (CLA)*, pp. 105-116, 2022.

[25] M. E. Cintra, M. C. Monard, H. A. Camargo, T. P. Martin, and A. Majidian, "On rule generation approaches for genetic Fuzzy Systems," in *Proceedings of the Congresso da Sociedade Brasileira de Computação*, 2011.

[26] T. J. Li and Y. Q. Wang, "Crisp-Fuzzy Concept Lattice Based on Interval-Valued Fuzzy Sets," *in Proceedings of the International Joint Conference on Rough Sets (IJCRS)*, vol. 14481, Springer, Cham, 2023, pp. 31. https://doi.org/10.1007/978-3-031-50959-9_31

[27] E. E. Shemis and A. M. Gadallah, "Enhanced Algorithms for Fuzzy Formal Concepts Analysis," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016 (AISI 2016)*, A. Hassanien, K. Shaalan, T. Gaber, A. Azar, and M. Tolba, Eds. Cham, Switzerland: Springer, 2017, vol. 533, Advances in Intelligent Systems and Computing, pp. 865-875. doi: 10.1007/978-3-319-48308-5_75.

[28] G. Fenza, M. Gallo, V. Loia, A. Petrone, and C. Stanzione, "Concept-drift detection index based on fuzzy formal concept analysis for fake news classifiers," *Technological Forecasting and Social Change*, vol. 194, p. 122640, 2023.

[29] M. Kelly, R. Longjohn, and K. Nottingham, "The UCI Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu. Accessed: May 2024.

[30] Y. Ke, J. Li, and S. Li, "Bit-Close: a fast incremental concept calculation method," *Applied Intelligence*, pp. 1-12, 2024.

[31] L. Zou, T. He, and J. Dai, "A new parallel algorithm for computing formal concepts based on two parallel stages," *Information Sciences*, vol. 586, pp. 514-524, 2022.

[32] S. Andrews, "Making use of empty intersections to improve the performance of CbO-type algorithms," in *Formal Concept Analysis: 14th International Conference, ICFCA 2017, Rennes, France,*

June 13-16, 2017, Proceedings 14, Springer International Publishing, 2017, pp. 56-71.

[33] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338-353, 1965. doi: 10.1016/S0019-9958(65)90241-X.

[34] A. Daghouri, K. Mansouri, and M. Qbadou, "Formal Concept Analysis based Framework for Evaluating Information System Success," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 4, 2019, doi: 10.14569/IJACSA.2019.0100451.

[35] M. Alwersh and L. Kovács, "K-Means Extensions for Clustering Categorical Data on Concept Lattice," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 9, 2023, doi: 10.14569/IJACSA.2023.0140953.

[36] S. Puri, "A Fuzzy Similarity Based Concept Mining Model for Text Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 11, 2011, doi: 10.14569/IJACSA.2011.021119