A New Efficient Approach for Updating Formal Fuzzy Concepts

Ebtesam Shemis
Computer science department
Faculty of Computers and Information, Menoufia University
Shebin el kom, Egypt
ORCID: 0000-0003-4781-926X
Email: ebtsam.elhoseni@ci.menofia.edu.eg

Gamal F. Elhady
Computer science department
Faculty of Computers and Information, Menoufia University
Shebin el kom, Egypt
Email: gamal.farouk@ci.menofia.edu.eg

Ammar Mohammed
Computer science department, Cairo university
Faculty of Graduate Studies for Statistical Research
Giza, Egypt
Email: ammar@cu.edu.eg

Arabi Keshk
Computer science department
Faculty of Computers and Information, Menoufia University
Shebin el kom, Egypt
Email: arabi.keshk@ci.menofia.edu.eg

***Abstract*— Formal fuzzy concept analysis is an effective data analysis and mining technique in the real world. However, deriving formal fuzzy concepts is an NP problem that demands substantial time and storage resources. With the continuous exponential growth of real-world data, there is a need to regularly update the extracted list of fuzzy concepts. This research paper presents a novel and efficient algorithm to update the extracted fuzzy concepts when inserting new data objects. The proposed algorithm eliminates the need for regenerating fuzzy formal concepts by reprocessing the entire dataset. Instead, it processes only the changed part and merges it with the old list of fuzzy concepts. We have evaluated the proposed approach over various datasets of different types: quantitative, categorical, and synthesized fuzzy data. The experimental results demonstrate that the proposed algorithm outperforms the traditional approach of fuzzy concept extraction by updating only the extracted fuzzy concepts rather than recreating them from scratch, especially in the case of massive data sets.**

Keywords—Fuzzy formal concept analysis, Fuzzy set theory, Dynamic fuzzy concepts, Dynamic real-world data, formal fuzzy concepts

I. Introduction

Formal concept analysis (FCA) is an effective technique for conceptual knowledge representation that supports mathematical modeling, analysis [1], and construction of conceptual hierarchies [2]. It is utilized to discover the related object groups and their attributes within a formal context (dataset) [3].

Classical FCA methodologies directly address a binary formal context, and as a result, they employ the crisp scaling technique (conceptual scaling) to manage contexts with multiple values. The crisp scaling method involves dividing the attribute domain into distinct intervals, where an object possesses a membership value of one for only one interval, while the remaining intervals have a membership value of zero [4]. The crisp scaling method requires excessive processing time and faces challenges in determining clear boundaries between scaled attributes. As it is difficult to decide where one attribute ends, and another begins. Fuzzy formal concept analysis (FFCA) has emerged as a solution to overcome the limitations of traditional FCA. FFCA can handle contexts with multiple values and fuzzy characteristics more logically and practically[5].

Today, real-world data is constantly changing as data objects are continuously inserted. Since extracting formal fuzzy concepts is a complex problem, it is promising to update the existing extracted formal fuzzy concepts when inserting new data objects instead of starting from scratch to regenerate the fuzzy concepts.

Several efforts propose algorithms to update crisp formal concepts and the formal concept lattice when inserting new binary data objects [6]. However, only a limited number of studies have addressed the challenge of updating formal fuzzy concepts when quantitative or multivalued data is introduced [7].

To our knowledge, the approach given by Zou et al [7] is the only fuzzy approach for updating the fuzzy concept list. However, it has limitations when it comes to updating fuzzy concepts upon inserting a new object. The current approach updates the concept list sequentially, attribute by attribute, but does not handle the update of fuzzy concepts when a new object is inserted. As a result, whenever a new object is inserted, the fuzzy concepts need to be regenerated from scratch.

Considering that data growth is more likely to occur in terms of objects rather than attributes, we propose a new approach to address this issue. Our approach focuses on handling changes in the fuzzy concept list in response to object insertion. Instead of regenerating fuzzy concepts from scratch, the proposed approach manages to block updates of the formal fuzzy concept list. It achieves this by simultaneously processing multiple objects and merging the resulting fuzzy formal concepts with the previously extracted ones. By adopting this approach, we aim to enhance the efficiency and effectiveness of updating the fuzzy concept list when new objects are inserted.

The structure of this paper is as follows: Section II presents the basic notions and formal definitions behind formal concept analysis (FCA) and fuzzy FCA. Besides, Section III explores related works for updating classical and fuzzy formal concepts when inserting new objects. Section IV introduces the proposed approach to handling block-based updates to the fuzzy concept list. Besides, Section V presents the experimental results and highlights the proposed approach's efficiency. Eventually, Section VI concludes the paper and presents future works.

II. PRELIMINARIES

This section revisits the basic notions and formal definitions of the classical formal concept analysis (FCA) and the corresponding notions and definitions of the fuzzy counterpart (fuzzy formal concept analysis, FFCA).

A. Formal Concept Analysis (FCA)

Formal concept analysis (FCA) is an applied mathematical discipline that relies on generating formal concepts and their hierarchical relationships. FCA offers a mathematical framework for representing, analyzing, and constructing conceptual knowledge structures [8].

FCA algorithms analyze datasets (formal contexts) and describe the relationship among objects and the corresponding attributes. These relationships can be described using formal concepts or a formal concept lattice [25].

In a formal context $\mathbb{K}(O, A, R)$, the dataset is structured as a binary table consisting of a set of rows (O) and a set of columns (A), along with an incident relation (R). The formal context serves as the primary input for FCA algorithms, which utilize it to generate formal concepts and construct a formal concept lattice.

Definition 1 A formal concept (X, Y) of formal context $\mathbb{K}(O, A, R)$ is a pair of object subset X and attribute subset Y such that $X \subseteq O$ is the formal concept extent and $Y \subseteq A$ is the concept intent. Besides, $X' = Y$ and $Y' = X$ must be satisfied such that X' and Y' is given by equations (1) and (2), respectively:

$$X' := \{a \in A \mid (x\ R\ a)\ \forall\, x \in X\} \qquad (1)$$

$$\mathrm{Y}' := \{\mathrm{o} \in \mathrm{O} \mid (\mathrm{o\ R\ y})\ \forall\, \mathrm{y} \in \mathrm{Y}\} \qquad (2)$$

Where X' gives a list of shared attributes in A that all objects in X have in common. As an alternative, X' can be written as $X \uparrow$.On the other hand, Y' gives all objects having all attributes in Y. An alternative notation to Y' is $Y \downarrow$. The notion $(x\ R\ a)$ formalizes that object x has the attribute a. Similarly, the notion $(o\ R\ y)$ formalizes that object o has the attribute y.

Formal concept lattice $\underline{\beta}$(O, A, R) visualizes all formal concepts extracted from a given context respecting the parent-child relationships given by equation (3). In this equation, A formal concept $(\mathrm{X}_1, \mathrm{Y}_1)$ is a sub-concept (child) of the concept $(\mathrm{X}_2, \mathrm{Y}_2)$ if and only if its extent (X_1) is a subset of (X_2).

$$(\mathrm{X}_1, \mathrm{Y}_1) \leq (\mathrm{X}_2, \mathrm{Y}_2) \Leftrightarrow \mathrm{X}_1 \subseteq \mathrm{X}_2 \Leftrightarrow \mathrm{Y}_2 \subseteq \mathrm{Y}_1 \qquad (3)$$

FCA algorithms classically take a binary formal context as their input. So, classical algorithms can't handle continuous quantitative data (multi-valued context MVC) directly. Instead, they need a crisp scaling method to preprocess. Crisp scaling maps the MVC context to an isomorphic binary context. To accomplish this, divide each quantitative attribute into a set of disjoint intervals [6]. Crisp scaling can convert the MVC context into a binary context, but it struggles with the challenge of defining distinct borders between scaled attributes. Furthermore, the mapping may result in misleading information [9, 10]. Fuzzy set theory [11] offered a solution to this problem and fuzzy formal concept analysis helps to effectively handle MVC contexts.

B. Fuzzy Formal Concept Analysis (FFCA)

The fuzzy set theory based FCA is an advanced variant of the traditional FCA that can efficiently handle a larger range of data types than the original FCA. FFCA has a wide range of real-world applications because it accommodates not only binary formal contexts but also quantitative and fuzzy formal contexts.

Many variants exist for FFCA, namely one-sided FFCA and full sided FFCA. One-sided FFCA only handles one side of the formal concept fuzzily while leaving the other side crisp. On the other hand, the full-sided FFCA employs fuzzy set theory in both the extent and intent sides of the formal concept. The single-sided FFCA is the focus of this paper's main contribution. So, this section revisits basic notions and formal definitions of single-sided FFCA. To dig deeper, readers can explore the following works [6, 12].

Definition 2: A formal fuzzy context $\widehat{K} = (O, A, R = \varphi\ (O \times A))$ consists of objects set O, attribute set A and a fuzzy incident relationship $R \in [0, 1]$ on the domain of $O \times A$ among objects and attributes. Each relation between object o and attribute a is denoted as $(o, a) \in R$ has a membership degree $\mu(o, a) \in [0,1]$. As an alternative notion, (o R a) represents the fact that object o has a relation to the attribute a with some extent degree between 0 and 1.

Definition 3: A formal fuzzy concept $(\widehat{X}, Y)$ of the fuzzy context $\widehat{K} = (O, A, R = \varphi\ (O \times A))$ consists of fuzzy extent $\widehat{X} \subseteq O$ and a crisp intent $Y \subseteq A$. To be a formal fuzzy concept the following conditions must hold: $\widehat{X}^{\uparrow} = Y$ and $Y^{\downarrow} = X$ where $X^{\uparrow}$and $Y^{\downarrow}$ are the derivation operators and are given by equations (4) and (5), respectively:

$$\widehat{X}^{\uparrow} := \{y \in Y \mid \forall x \in \widehat{X}:\ \mu_R(x, y) \geq \mu_{\widehat{X}}(x)\} \qquad (4)$$

$$Y^{\downarrow} := \left\{\frac{\mu_{\widehat{X}}(x)}{x} \middle|\ \mu_{\widehat{X}}(x) = min_{y \in Y}\ (\mu_R(x, y))\right\} \qquad (5)$$

Given that $\widehat{X}^{\uparrow}$ and $Y^{\downarrow}$ are the concept crisp intent and fuzzy extent, respectively. Another common notion to $\widehat{X}^{\uparrow}$ and $Y^{\downarrow}$ are $\widehat{X}'$ and Y', respectively.

For more clarification, the following *Example 1* illustrates a sample fuzzy context and the process of extracting fuzzy formal concepts respecting the formal definitions given in Definition (3).

Example 1: Given an example fuzzy context (*TABLE I)* with four objects $O = \{0, 1, 2, 3\}$ and five fuzzy attributes A= $\{a, b, c, d, e\}$

TABLE I AN EXAMPLE FUZZY FORMAL CONTEXT

ID	*a*	*b*	*c*	*d*	*e*
0	0.6	1	0.5	0	0
1	0.7	1	0.6	0	0
2	1	1	0.9	0	1
3	0.5	0.4	0	0	0

Following definition 3, TABLE *II* shows the entire set of fuzzy concepts extracted from the example fuzzy context in *TABLE I.* As shown, each single-sided fuzzy concept comprises a fuzzy set representing the concept extent and a crisp set representing the concept intent. The fuzzy extent of the concept consists of the object ID in the nominator and the corresponding membership degree in the dominator $(\frac{o_{id}}{\mu_{o_{id}}})$.

TABLE II. FUZZY CONCEPTS EXTRACTED FROM THE FUZZY CONTEXT IN TABLE I

#	**Single-Sided Fuzzy Formal Concept**	
	Fuzzy concept extent	***Crisp concept intent***
0	$\{\frac{0}{1.0}, \frac{1}{1.0}, \frac{2}{1.0}, \frac{3}{1.0}\}$	[ ]
1	$\{\frac{0}{0.6}, \frac{1}{0.7}, \frac{2}{1.0}, \frac{3}{0.5}\}$	$[a]$
2	$\{\frac{0}{0.6}, \frac{1}{0.7}, \frac{2}{1.0}, \frac{3}{0.4}\}$	$[a, b]$
3	$\{\frac{0}{0.5}, \frac{1}{0.6}, \frac{2}{0.9}\}$	$[a, b, c]$
4	{ }	$[a, b, c, d, e]$
5	$\{\frac{2}{0.9}\}$	$[a, b, c, e]$
6	$\{\frac{2}{1.0}\}$	$[a, b, e]$
7	$\{\frac{0}{1.0}, \frac{1}{1.0}, \frac{2}{1.0}, \frac{3}{0.4}\}$	$[b]$

Equations (4) and (5) demonstrate the extraction of each fuzzy concept. For example, let us consider the fuzzy concept numbered two $(\{\frac{0}{0.6}, \frac{1}{0.7}, \frac{2}{1.0}, \frac{3}{0.4}\}, [a, b])$ in Table II, whose concept intent is $[a, b]$. As per equation (5), the fuzzy extent $[a, b] \downarrow$ is the minimum membership value for each object having attributes a and b (highlighted in the fuzzy context). The minimum membership for attributes a and b is 0.6 for object 0, 0.7 for object 1, 1.0 for object 2, and 0.4 for object 3. Consequently, the resulting fuzzy extent of this concept is the set of all these objects and their membership values $(\{\frac{0}{0.6}, \frac{1}{0.7}, \frac{2}{1.0}, \frac{3}{0.4}\})$.

Definition 4: The fuzzy concept $(\widehat{X_1}, Y_1)$ is a sub-concept of the fuzzy concept $(\widehat{X_2}, Y_2)$ if and only if $\widehat{X_1} \subseteq \widehat{X_2} \Leftrightarrow Y_2 \subseteq Y_1$. Such a relation is denoted by $(\widehat{X_1}, Y_1) \preccurlyeq (\widehat{X_2}, Y_2)$.

A fuzzy concept lattice visualizes the complete set of fuzzy concepts extracted and ordered by the sub-concept relationship ($\preccurlyeq$).

III. PREVIOUS WORKS

This section explores classical and fuzzy related approaches and algorithms that effectively address updating the pre-existing formal concepts or concept lattice when inserting new data. This section covers both crisp-based works and fuzzy-based ones.

Algorithms for generating formal concepts lie in one of the following two main categories: 1) Batch algorithms: process the formal context once and produce the formal concepts' list. 2) Incremental algorithms that incrementally process the formal context, one object or attribute at a time [13].

The batch algorithms exhibit a faster identification of concepts compared to the incremental algorithms due to the incorporation of a canonicity test, which effectively prevents the redundant checking of the same concept [14].

Batch algorithms extract formal concepts in a top-down (or bottom-up) method but cannot be used when new data is inserted into the formal context (they are not updatable). Examples of batch algorithms are PCbO [15], FCbO [16], and parallel InClose algorithm [17].

Incremental algorithms are well-suited for constructing the formal concept lattice and managing newly inserted data. They process the formal context object by object or attribute by attribute. Consequently, when adding a new object, incremental algorithms update the extracted formal concept list or lattice accordingly. While these algorithms are suitable for dynamic data changes, they require additional processing time [6, 14].

A. Existing Algorithms for Updating Formal Concepts

In the literature, there are many remarkable classical algorithms for incrementally generating formal concepts. Those classical algorithms handle only binary contexts. A widely acknowledged update algorithm is the AddIntent algorithm [24] which laid the foundation of incremental update algorithms.

The AddIntent algorithm updates the concept lattice by adding one object at a time. It handles only binary formal contexts and starts processing each object and updates the lattice accordingly. The AddIntent algorithm may revisit concepts unnecessarily due to the presence of shared parent nodes in the lattice structure. This algorithm excels in dense datasets with high fill ratio, but degrades efficiency in randomized, sparse datasets with small fill ratios.

Enhanced versions of AddIntent include the Improved AddIntnet [18], FastAddIntent [19] and Zhi et al [20] algorithms. These algorithms aim to reduce the unnecessary revisits to concepts of the original AddIntnet algorithm to make it more time efficient.

The Improved AddIntent algorithm [18] utilizes a lattice structure to prevent unnecessary revision of concepts. It achieves this by marking concepts as visited, new, or modified, and considering their sub-concept relationships. So, it can bypass redundant comparisons of upper concepts. The Improved AddIntent algorithm reduces computation time compared to the AddIntent algorithm, especially for formal contexts with more attributes than objects. But it requires additional labels per concept, consuming more memory. Besides, it slightly surpasses AddIntent in cases with more objects [19].

The FastAddIntent [19] algorithm improves the efficiency of the AddIntent algorithm by enhancing how it finds new concepts and canonical generators, which are the most time-consuming processes. The FastAddIntent algorithm best fits large contexts with large objects in practical datasets.

Zhi and Li [20] discovered that when a formal concept is modified, its upper neighbors are either modified or entirely new concepts. They created a queue, focusing on exploring only modified or new concepts, reducing search space compared to the original AddIntent algorithm. Zhi et al.'s approach not only updates the concept lattice when new objects are added but also updates the association rules that have already been discovered based on the updated lattice.

The FastAddExtent algorithm [21] updates the concept lattice when inserting new attributes into the formal context. But it does not handle the lattice update when new objects are inserted. The FastAddExtent algorithm, like the Improved AddIntent algorithm, demonstrates significantly superior performance when there are more attributes than objects. In terms of processing time, the FastAddExtent algorithm has surpassed the improved AddIntent algorithm, although it requires more memory due to the utilization of four data fields for each concept in the lattice.

B. *Analysis of existing algorithms*

This section analyzes the limitations inherent in the related approaches while investigating the differences between related works and the proposed approach.

TABLE III presents a comprehensive summary of the relevant literature, encompassing the respective contributions, advantages, and limitations. Notably, AddIntent [18], FastAddIntent [19], Zhi et al [20], and FastAddExtent [21] represent classical approaches that exclusively cater to binary formal contexts, so they are incapable of accommodating diverse other data types directly, such as fuzzy and quantitative data. In contrast, our proposed algorithm is founded on a fuzzy-based framework that effectively handles the entire spectrum of data types, encompassing both fuzzy and quantitative representations.

Fuzzy approaches for updating formal concepts are very rare [6]. To our knowledge only Zou et al. [7] developed an impressive algorithm that effectively handles fuzzy context. The approach of Zou et al. generates fuzzy concepts by incrementally processing each attribute. Rather than completely regenerating the fuzzy concepts, they update the existing fuzzy concepts when new attributes are added one at a time. However, the fuzzy concepts are re-extracted from scratch when adding new objects. On the contrary, the proposed approach updates fuzzy concepts when adding new objects as a group without the need to regenerate the concepts from the beginning. In summary, the proposed approach is unique in its type that:

1. Handles all types of data (which classical approaches [18–21, 24] can't handle).
2. Update *Fuzzy* formal concepts upon inserting objects,
3. Process a group of objects once without instant updates, one by one.

TABLE III ANALYSIS OF EXISTING ALGORITHMS

Classical approaches that handle binary contexts		
Approach	***Contribution and advantages***	***Limitation***
AddIntent [24]	• Removes unneeded checks by disregarding unaffected concepts. • Best fits real dense dataset	• Sparse dataset inefficiency. • Unnecessary concept revisiting. • Update the complete lattice for every newly inserted object. • *Doesn't handle Fuzzy data.*
Improved AddIntent [18]	• Reducing concept revisits than AddIntent algorithm. • More efficient than AddIntent • Performs well with more attributes.	• Low efficiency when |objects| < |attributes| • High memory usage. • Update the complete lattice for every newly added object. • *Doesn't handle Fuzzy data.*
Fast Add Intent [19]	• Enhancing canonical generators search and concept link recovery. • Ideal for large contexts with more objects. • More efficient than AddIntent.	• Sparse dataset inefficiency. • Update the complete lattice for every newly added object. • *Doesn't handle Fuzzy data.*
Zhi and Li [20]	• Reduces AddIntent algorithm overhead by utilizing a modified concept queue, limiting search to relevant concepts only. • Speeds up search by shrinking the concept search space.	• Update the complete lattice for every newly added object. • *Doesn't handle Fuzzy data.*
Fast Add Extent [21]	• Update the lattice when inserting new attributes, eliminate superfluous concept comparisons. • Efficiency improved with increased attributes.	• Update the complete lattice for every newly added attribute. • Significant memory consumption
Fuzzy approaches that handle fuzzy or quantitative contexts		
Approach	***Contribution and advantages***	***Limitation***
Zou et al. [7]	• Handles all data types. • Update fuzzy concepts when inserting new attributes.	• Process newly added attributes one at a time. • Doesn't handle updating concepts when new objects.

IV. PROPOSED APPROACH

This section presents the proposed approach for updating the already extracted formal concepts when inserting new rows (objects) into the dataset.

Fig. 1 illustrates the methodology of the proposed approach. Initially, the proposed approach involves processing the fuzzy formal context once to generate one-sided concepts. These fuzzy concepts are then saved on the disk. When new data is added, only the inserted data is processed to extract its fuzzy concepts. Subsequently, the $MergeFuzzyConcepts$ algorithm combines the original stored fuzzy concepts with the newly generated ones.

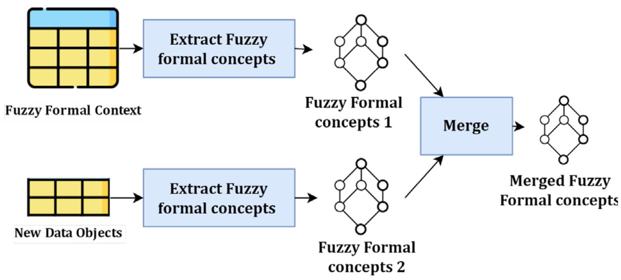

Fig. 1 The proposed approach

The proposed approach represents a pioneering method for processing the inserted data as a cohesive unit rather than incrementally updating the extracted fuzzy concepts for each object. Instead, it focuses solely on the new portion as a compact fuzzy context and subsequently integrates it with the original fuzzy concepts through a merging process.

The proposed approach utilizes a batch algorithm to generate fuzzy concepts for the original dataset and the new data. We use the improved version of fuzzy CbO [22] due to its efficiency, as batch algorithms are faster than incremental ones.

Proposed Algorithm: MergeFuzzyConcepts (A, B)

Input: Two fuzzy concepts' lists (A and B) with the same attribute sets.
Output: Merged Fuzzy Formal Concepts List ($\mathbf{C_{new}}$)
Steps:

```
C_new ← [ϕ]
Foreach c_1 ∈ A:
    Foreach c_2 ∈ B:
        Intent_c ← {c_1.intent ∩ c_2.intent }
        If C_new ⊉ Intent_c then,
            Extent_c ← s_norm (c_1.extent, c_2.extent)
            C_new ← C_new ∪ {Extent_c, Intent_c}
        end if
    End for
End for
```

The MergeFuzzyConcepts algorithm inputs two lists of fuzzy concepts, A and B. These lists represent fuzzy formal concepts with the same attribute sets. The algorithm aims to merge these concepts into a new fuzzy concepts list, namely C_{new}. To achieve this, the algorithm iterates over each concept in A and B. For each pair of concepts (c_1 from A and c_2 from B), it calculates the intersection of their intents, denoted as $Intent_c$. If the intersected intent ($Intent_c$) is a new intent, the algorithm proceeds to calculate the fuzzy extent of the merged concept, $Extent_c$. This is done using a specific fuzzy set union operator (s_norm). The s_norm operator (t-conorm) is the classical Zadeh's fuzzy union operator, which is defined in Definition (5).

Definition 5: Let X and Y be two fuzzy sets defined on the same universe of discourse Z. The fuzzy union of X and Y, denoted as $X \cup Y$, is a fuzzy set defined on Z, where the membership function $\mu_{X \cup Y}(z)$ represents the degree of membership of an element $z \in Z$ in the fuzzy set $X \cup Y$. The membership function $\mu_{X \cup Y}(z)$ is calculated using the maximum operator (max) as follows:

$$\mu_{X \cup Y}(z) = max(\mu_X(z), \mu_Y(z)) \tag{4}$$

where $\mu_X(z)$, and $\mu_Y(z)$ are the membership values of z in fuzzy sets X and Y, respectively.

Example 2: Consider TABLE IV, which represents the new objects to be inserted into the previous fuzzy context shown in TABLE I. Following the proposed algorithm, the proposed approach utilizes the previously generated fuzzy concepts in TABLE II, namely c_1 . Subsequently, the approach generates the fuzzy concepts of the newly inserted rows, namely c_2, as illustrated in TABLE V. The next step involves merging the fuzzy concepts' lists, c_1 and c_2, into a new list, namely C_{new}, which is then stored. The result of applying the proposed algorithm ($MergeFuzzyConcepts$) to the fuzzy concepts c_1 and c_2 in TABLE II and TABLE V, respectively, is presented in TABLE *VI*.

In TABLE *VI*, the fuzzy concepts with numerical labels {0, 1, 2, 3, 4, 5, 6} have been modified by updating only their fuzzy extent, as their intent appeared previously in the original concepts list in TABLE *II*. On the other hand, the fuzzy concepts numbered {8, 9} are newly created concepts resulting from the merging process of the new inserted objects.

TABLE IV NEW DATA TO BE INSERTED TO THE FUZZY CONTEXT IN TABLE I

ID	*a*	*b*	*c*	*d*	*e*
4	0.5	0.5	1	0	0
5	0.6	0.9	0.9	0	0

TABLE V FUZZY CONCEPTS EXTRACTED FROM THE FUZZY CONTEXT IN TABLE IV

#	Single-Sided Fuzzy Formal Concept	
	Fuzzy concept extent	***Crisp concept intent***
0	$\{\frac{4}{1.0}, \frac{5}{1.0}\}$	[ ]
1	$\{\frac{4}{0.5}, \frac{5}{0.6}\}$	$[a, b, c]$
2	{ }	$[a, b, c, d, e]$
3	$\{\frac{4}{0.5}, \frac{5}{0.9}\}$	$[b, c]$
4	$\{\frac{4}{1.0}, \frac{5}{0.9}\}$	$[c]$

TABLE VI MERGED FUZZY CONCEPTS FROM FUZZY CONCEPTS IN TABLE II AND FUZZY CONCEPTS IN TABLE V

#	Single-Sided Fuzzy Formal Concept	
	Fuzzy concept extent	***Crisp concept intent***
0	$\left\{\frac{0}{1.0}, \frac{1}{1.0}, \frac{2}{1.0}, \frac{3}{1.0}, \frac{4}{1.0}, \frac{5}{1.0}\right\}$	[ ]
1	$\left\{\frac{0}{0.6}, \frac{1}{0.7}, \frac{2}{1.0}, \frac{3}{0.5}, \frac{4}{0.5}, \frac{5}{0.6}\right\}$	$[a]$
2	$\left\{\frac{0}{0.6}, \frac{1}{0.7}, \frac{2}{1.0}, \frac{3}{0.4}, \frac{4}{0.5}, \frac{5}{0.6}\right\}$	$[a, b]$
3	$\left\{\frac{0}{0.5}, \frac{1}{0.6}, \frac{2}{0.9}, \frac{4}{0.5}, \frac{5}{0.6}\right\}$	$[a, b, c]$
4	{ }	$[a, b, c, d, e]$
5	$\left\{\frac{2}{0.9}\right\}$	$[a, b, c, e]$
6	$\left\{\frac{2}{1.0}\right\}$	$[a, b, e]$
7	$\left\{\frac{0}{1.0}, \frac{1}{1.0}, \frac{2}{1.0}, \frac{3}{0.4}, \frac{4}{0.5}, \frac{5}{0.9}\right\}$	$[b]$
8	$\left\{\frac{0}{0.5}, \frac{1}{0.6}, \frac{2}{0.9}, \frac{4}{0.5}, \frac{5}{0.9}\right\}$	$[b, c]$
9	$\left\{\frac{0}{0.5}, \frac{1}{0.6}, \frac{2}{0.9}, \frac{4}{1.0}, \frac{5}{0.9}\right\}$	$[c]$

V. EXPERIMENTAL RESULTS

This section visualizes experiments performed over the classical way of generating fuzzy concepts versus the proposed approach that manages only the update of the newly inserted data.

The proposed approach is different from other approaches of its kind and cannot be directly compared to them. TABLE VII provides further details on why such comparisons are infeasible. As demonstrated, all studies [18–21, 24] rely on crisp sets of both concept intent and extent. These approaches are only capable of operating in binary formal contexts where values of either 0 or 1 are allowed. In contrast, the proposed approach operates over fuzzy contexts where data represents the membership of an object to an attribute subset within the range of [0, 1].

Although Zou et al.'s work [7] adopts a fuzzy approach, it introduces an incremental algorithm that updates the fuzzy lattice by inserting new attributes one by one. When updating objects, their approach will generate the concept lattice from scratch. On the contrary, the proposed approach handles the update of the fuzzy concepts when inserting new objects. Due to the fundamental differences in the update process, conducting an experimental comparison between this approach and the proposed approach is not feasible. However, experiments in this section show the benefits of updating fuzzy concepts using the proposed approach when new data objects are inserted versus the traditional generation of fuzzy concepts from scratch.

TABLE VII CHARACTERISTICS OF THE PROPOSED APPROACH VERSUS RELATED WORKS

Approach	Data Supported	Update Is Based on Inserting	Update Methodology
AddIntent [24]	Binary	New objects	One by one update
Improved AddIntent [18]			
FastAddIntent [19]			
Zhi and Li [20]			
FastAddExtent [21]			
Zou et al. [7]	***Fuzzy***	New Attributes	
Proposed Approach		***New objects***	***Block based update***

TABLE *VIII* shows the datasets that are used in the experiments. Diverse types of data are used to examine the approach with heterogeneous data. For instance, both abalone and Iris datasets are multivariate with different attribute types. They are available on the UCI ML repository [23].

TABLE VIII DATASETS USED FOR EXPERIMENTS

Dataset	Density (fill ratio)	Description	(Total objects, Total attributes
Abalone	70%	Multivariate (Categorical, Integer, Real)	(4177, 19)
Iris	58%	Multivariate (Categorical, Real)	(150, 11)
Random Fuzzy_10,000	20%, 30%	Synthetic Fuzzy Data	(10,000, 10)

To fuzzify the quantitative attributes, we use three linguistic labels for each quantitative attribute (low, middle, and high), such that both low and high linguistic labels are defined as triangular membership functions. Besides, the middle linguistic label is defined using a trapezoidal membership function. The number of attributes shown in TABLE *VIII* are the ones produced after fuzzification and the density ratio is calculated after fuzzification. Besides, we generated an artificial fuzzy dataset, namely, Random Fuzzy_10000, with different density ratios of (20% and 30%) to evaluate the proposed approach over a sparse and dense fuzzy dataset.

We examined the abovementioned datasets by generating the original set of fuzzy concepts, $fuzzyConcepts_1$, for a subset of objects O_A in the dataset and then storing it. Afterwards, we insert another subset of objects O_B to see how long it takes to produce and merge with the originally generated fuzzy concepts, $fuzzyConcepts_1$. Finally, we compare the time it takes to update the fuzzy concepts with the time it takes to construct the complete set of fuzzy concepts from scratch (for all objects $O_A \cup O_B$).

We coded all algorithms utilized in the suggested methodology using the Python programming language and executed them on a Windows 10 environment, specifically on an Intel Core i5 2.30GHz machine with 8 GB of RAM.

Fig. 2 visualizes the experiment results for the abalone dataset. Initially, the first sub-context has the first 1000 objects of the original dataset, and each run involves increasing the number of objects by 50. The experiment result shows how the proposed approach produces the same output as generating the fuzzy concepts from scratch but with lower processing time.

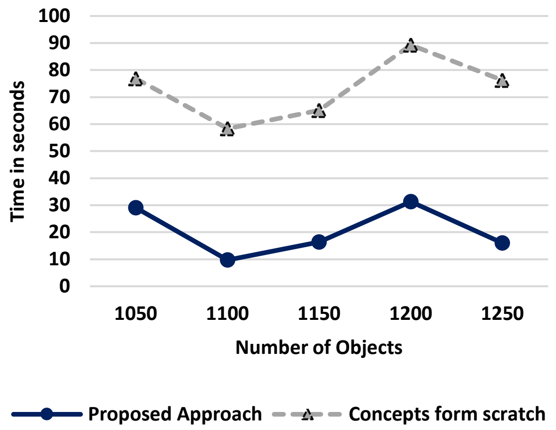

Fig. 2 Proposed approach versus extracting concepts from scratch over abalone dataset

Fig. 3 visualizes the experiment results for the fuzzy Iris dataset. Initially, the first sub-context has the first one hundred objects of the original dataset, and each run involves increasing the number of objects by ten until reaching the entire 150 objects of the dataset. The experiment result over fuzzy iris shows the efficiency of the proposed approach over generating fuzzy concepts from scratch.

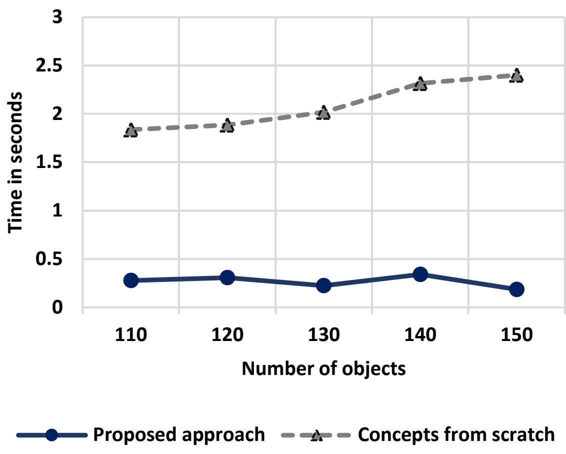

Fig. 3 Proposed approach versus extracting concepts from scratch over Fuzzy Iris dataset

We have evaluated the proposed approach over the synthetic fuzzy dataset (Random Fuzzy_10000) with 20% density ratio. The experiment result is depicted in Fig. 4. The initial sub-context started with 7500 objects, and each run involved increasing the number of objects by five hundred. Again, the proposed approach proves its efficiency versus generating fuzzy concepts from scratch.

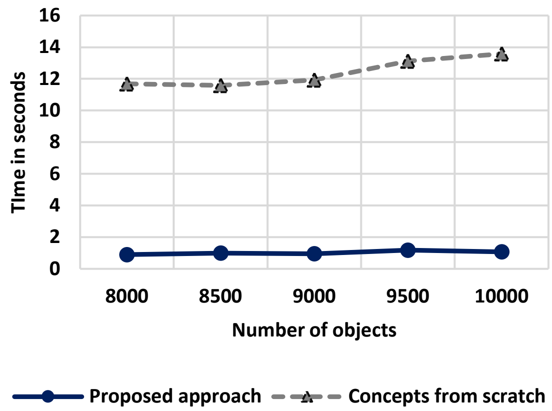

Fig. 4 Proposed approach versus extracting concepts from scratch over the synthetic fuzzy dataset

To demonstrate the proposed approach's ability to handle fuzzy datasets with various density ratios, we have conducted supplementary experiments to assess its performance over sparse and dense datasets. TABLE *IX* illustrates the experiments results over the Fuzzy_10000 dataset with density ratios of 20% and 30%.

In TABLE *IX*, $part_1$ demostrate the data part that have previously been processed and from which fuzzy concepts are extracted and stored. Additionally, the table presents a comparison of the processing time required to extract the same number of fuzzy concepts using both the proposed approach and the classical approach. It is evident that as the density increases, the performance of both the proposed approach and the classical approach of extracting fuzzy concepts from scratch decreases. However, the proposed approach still outperforms the classical approach by a considerable margin.

TABLE IX EXPERIMENTS OVER FUZZY CONTEXTS WITH DIFFERENT DENSITY RATIOS

Density	$Part_1$	Total	No. of Fuzzy concepts	Proposed Approach (s)	Classical extraction (s)
20%	7500	8000	375	0.90	10.79
	8000	8500	382	0.99	10.60
	8500	9000	383	0.96	10.98
	9000	9500	388	1.18	11.93
	9500	10000	396	1.07	12.51
30%	7500	8000	925	6.36	27.03
	8000	8500	936	6.42	26.68
	8500	9000	941	5.41	28.29
	9000	9500	942	6.83	29.77
	9500	10000	942	5.22	33.75

VI. CONCLUSIONS AND FUTURE WORKS

This paper introduces a novel and efficient algorithm for updating extracted fuzzy concepts in fuzzy formal concept analysis. By eliminating the need to regenerate fuzzy concepts from scratch and instead processing only the changed part and merging it with the existing list, the proposed algorithm offers significant advantages over the traditional fuzzy approach. The experimental results demonstrate that the algorithm outperforms the traditional method, particularly when dealing with sparse data sets. This advancement in updating fuzzy concepts addresses the challenge posed by the continuous exponential growth of real-world data and provides a valuable contribution to data analysis and mining techniques.

As future work, we plan to utilize the proposed algorithm in parallel computing of fuzzy concepts. This can be accomplished by dividing the dataset into segments according to objects and processing them in parallel. Finally, we can merge the resulting fuzzy concepts using the proposed algorithm.

REFERENCES

[1]. L. Kwuida and R. Missaoui, "Formal Concept Analysis and Extensions for Complex Data Analytics," in Complex Data Analytics with Formal Concept Analysis, R. Missaoui, L. Kwuida, and T. Abdessalem, Eds., ed Cham: Springer International Publishing, pp. 1-15, 2022.

[2]. L. Zhang and Y. Jiang, "Fusing semantic aspects for formal concept analysis using knowledge graphs," Multimedia Tools and Applications, pp. 1-35, 2023.

[3]. G. Lang and Y. Yao, "Formal concept analysis perspectives on three-way conflict analysis," International Journal of Approximate Reasoning, vol. 152, pp. 160-182, 2023.

[4]. A. Izadpanahi, "Relational Approach to the L-Fuzzy Concept Analysis, Msc thesis," ed: Brock University, 2022.

[5]. H. Zhi, J. Li, and Y. Li, "Multilevel Conflict Analysis Based on Fuzzy Formal Contexts," IEEE Transactions on Fuzzy Systems, vol. 30, no. 12, pp. 5128-5142, 2022.

[6]. 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, p. e1401, 2021.

[7]. C. Zou, H. Deng, J. Wan, Z. Wang, and P. Deng, "Mining and updating association rules based on fuzzy concept lattice," Future Generation Computer Systems, vol. 82, pp. 698-706, November 2017.

[8]. S. Ferré, M. Huchard, M. Kaytoue, S. O. Kuznetsov, and A. Napoli, "Formal Concept Analysis: From Knowledge Discovery to Knowledge Processing," in A Guided Tour of Artificial Intelligence Research: Volume II: AI Algorithms, P. Marquis, O. Papini, and H. Prade, Eds., ed Cham: Springer International Publishing, pp. 411-445, 2020

[9]. V. A. Semenova and S. V. Smirnov, "Extended methodology for deriving formal concepts," Journal of Physics: Conference Series, vol. 2099, p. 012026, 2021.

[10]. S. Zheng, Y. Zhou, and T. Martin, "A New Method for Fuzzy Formal Concept Analysis," in 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 405-408, 2009.

[11]. L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, pp. 338-353, 1965.

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

[13]. Y. Dong, Y. Wu, and Z. Liu, "Research on Two Main Construction Methods of Concept Lattices," Journal of Shanghai Jiaotong University (Science), vol. 24, pp. 243-253, 2019.

[14]. R. K. Chunduri and A. K. Cherukuri, "Scalable formal concept analysis algorithms for large datasets using spark," Journal of Ambient Intelligence and Humanized Computing, vol. 10, pp. 4283-4303, 2019.

[15]. P. Krajca, J. Outrata, and V. Vychodil, "Advances in Algorithms Based on CbO," in CLA, vol. 672, pp. 325-337, 2010.

[16]. J. Outrata and V. Vychodil, "Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data," Information Sciences, vol. 185, pp. 114-127, 2012.

[17]. N. Kodagoda, S. Andrews, and K. Pulasinghe, "A parallel version of the in-close algorithm," in 2017 6th National Conference on Technology and Management (NCTM), pp. 1-5, 2017.

[18]. L. Lingling, Z. Lei, Z. Anfu, and Z. Funa, "An improved addintent algorithm for building concept lattice," in 2011 2nd International Conference on Intelligent Control and Information Processing, pp. 161-165, 2011.

[19]. L. Zou, Z. Zhang, and J. Long, "A fast incremental algorithm for constructing concept lattices," Expert Systems with Applications, vol. 42, pp. 4474-4481, 2015.

[20]. H. Zhi and J. Li, "Influence of dynamical changes on concept lattice and implication rules," International Journal of Machine Learning and Cybernetics, vol. 9, pp. 795-805, May 2018.

[21]. J. Zhang, R. Liu, L. Zou, and L. Zeng, "A New Rapid Incremental Algorithm for Constructing Concept Lattices," Information, vol. 10, p. 78, 2019.

[22]. 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, Advances in Intelligent Systems and Computing, vol 533. Springer, Cham, pp. 781-792, 2017.

[23]. A. Asuncion and D. Newman, "UCI machine learning repository, https://archive.ics.uci.edu", 2007, last visited 20-Jul-2023.

[24]. D. Van Der Merwe, S. Obiedkov, and D. Kourie, "Addintent: A new incremental algorithm for constructing concept lattices," in International Conference on Formal Concept Analysis, pp. 372-385, 2004.

[25]. E. Shemis, G. F. Elhady, A. Mohammed and A. Keshk, "A Fuzzy-Crisp Frequent Concept Lattice Generation Algorithm" in 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 75-80, 2022