University of Sfax
National Engineering School of Sfax
Doctoral school Sciences & Technologies

Universidad de Las Palmas de Gran Canaria
Programa de Doctorado en Empresa, Internet y Tecnologías de las Comunicaciones

DOCTORAL THESIS

submitted in the framework of a cotutelle agreement
for the joint award of the degree of:

Doctorat Degree of University of Sfax, Tunisia

Computer System Engineering

&

Título de Doctor de la Universidad de Las Palmas de Gran Canaria, España

Sociedad, Empresa y Tecnologías de la Información y el Conocimiento

Hybrid Bio-inspired Ant Colony Clustering approach for Constructing Collaborative Learning Teams

Enfoque de agrupamiento híbrido basado en inteligencia de colonias de hormigas para la creación de equipos colaborativos

Presented By:

ABIR ABID

Under the supervision of:

- **Prof. Mounir Ben Ayed**, ReGIM-Lab,ENIS, University of Sfax, Tunisia
- **Prof. Ilhem Kallel**, ReGIM-Lab, ENIS, University of Sfax, Tunisia
- **Prof. Javier Sanchez Medina**, CICEI, University of Las Palmas Gran Canaria, Spain

Defense Date: [Day Month Year]

Defense Location: Las Palmas De Gran Canaria, Spain

Academic Year: 2025–2026

Dedication

This thesis is dedicated with profound love and gratitude to my cherished family, whose unwavering support, encouragement, and belief in me have made this journey possible.

To my beloved family,
Whose love, patience, and faith have been the silent strength behind every step of this journey.

To my friends and mentors,
For their encouragement, kindness, and belief in me, even in moments of doubt.

To my younger self,
Who dared to dream, even when the path seemed uncertain;
Who faced doubts and fears with quiet courage;
And who never stopped believing in the power of knowledge, growth, and perseverance.

"The future belongs to those who believe in the beauty of their dreams"-Eleanor Roosevelt

Acknowledgements

I express my deepest gratitude to all those who contributed to this work; without their support and guidance, this thesis would not have been possible.

First and foremost, I am deeply grateful to **Prof. Mounir Ben Ayed**. His invaluable guidance, insightful advice, and constructive comments have greatly shaped the direction and quality of the research presented in this thesis.

I extend my deepest and heartfelt gratitude to **Prof. Ilhem Kallel**, whose exceptional guidance has been indispensable throughout this journey. Her deep scientific expertise, insightful advice, and meticulous attention to detail have greatly shaped the quality and direction of this thesis. Beyond her invaluable academic support, she has always encouraged me morally, providing motivation, patience, and confidence in my abilities during the most challenging moments. Her dedication, kindness and unwavering support have been a constant source of inspiration and I feel truly privileged to have had the opportunity to work under her mentorship.

I also sincerely thank **Prof. Javier Sánchez Medina** for his thoughtful guidance and valuable insights that significantly enriched this research.

I extend my sincere gratitude to **Prof. Habib M. Kammoun**, head of the Computer Science and Communications Department at the Faculty of Science of Sfax, for his unwavering support and continuous encouragement throughout this journey. His leadership and dedication to excellence have been an enduring source of inspiration, contributing significantly to the realization of this work.

I am grateful to **Prof. [Jury: tba]** for the honor of chairing my thesis committee and for his constructive feedback and support during the evaluation process.

My sincere appreciation goes to **Prof. [Jury: tba]** for their careful review, insightful comments, and valuable suggestions that contributed to enhancing this thesis.

I also thank **Prof. [Jury: tba]** for generously agreeing to examine this work and for providing his expertise.

Contents

List of Figures

List of Tables

Introduction

Collaborative learning is an umbrella term for a wide variety of educational forms. It encourages students working in groups of two or more , to explore and share resources, knowledge, solutions, ideas, and thoughts to solve problems.

Unlike traditional learning methods, which are based on individual learning performance Abid et al. (2024), collaborative learning helps learners at various performance levels work together in small groups toward a common goal. As a result, each member is accountable for both their peers' learning and their own. Collaborative learning practitioners emphasize that this approach is fundamentally about building learning communities. It aims not only to improve learners' academic skills but also to develop their interpersonal skills Asad and Qureshi (2025) Männistö et al. (2020).

However, one of the key challenges in collaborative learning is creating groups where, learners need to feel safe, comfortable, when assigned to a suitable team. Therefore, forming the appropriate groups is one of the fundamental pillars of the collaborative learning.

This issue has inspired many researchers in the educational field area, who apply different algorithms to address the grouping problem and discover optimal learner groups. Some researchers rely on computational approaches Chen and Li (2024), while others draw on heuristic ideas inspired by entomological studies of living organisms in nature Kiran et al. (2022).

Nature-inspired meta-heuristic algorithms, developed based on principles drawn from biological evolution. They are well known in machine learning for addressing optimal solutions of complex problems.

Numerous bio-inspired algorithms exist in the literature, such as Genetic Algorithms (GA) Forrest (1996), Particle Swarm Optimization (PSO) Kennedy and Eberhart (1995); Shami et al. (2022), Ant Colony Optimization (ACO) Dorigo and Di Caro (1999), Bee

Colony Optimization (BCO) Teodorović et al. (2022), etc. These algorithms have been successfully adapted and have found success in solving clustering and NP-complete problems.

Among several bio-inspired collaborative systems, ant-based clustering techniques have proven successful in solving clustering problems. These methods are inspired by the ecological study of social ants. Being talking about social, means that ants cannot survive on their own since they belong to the social category. Therefore, they demonstrate remarkable coordination of activities among colony members.
In recent years, ant-based clustering techniques received special attention from the research community due to their performance in exploratory data analysis across many fields, including collaborative robotics as in (Kallel et al. (2008), Chatty et al. (2011); Chatty et al. (2012); Chatty et al. (2013)). However, further investigation is needed to address issues related to performance, convergence, robustness, stability, etc.

Our research journey revealed two main challenges. On one hand, bio-inspired algorithms involve different parameters which is significantly impact their performance. Parameter tuning (or setting) has attracted considerable research attention over the several researchers in the past decade Nannen and Eiben (2006) as even though the algorithm is efficient, setting inappropriate values of parameters may lead to a low-quality solutions. For example, in Hassanat et al. (2019), the authors stated that the integration among parameters such as mutation, crossover rates, and population size is vital for successful GA search. Similarly, the searching capability of the PSO algorithm is directly influenced by its three main control parameters (inertia weight, cognitive acceleration coefficient, and social acceleration coefficient). In fact, as noted by Carlisle and Dozier (2001), Trelea (2003) and Van den Bergh and Engelbrecht (2006), a priory tuning of PSO control parameters can significantly improve performance but remains highly sensitive to these settings.

Likewise, ant-based clustering-inspired by larval sorting activities and corpses clustering observed in real ant colonies, the ant based clustering has emerged as a new method for solving clustering problems. Just like other bio-inspired algorithms, ant colony algorithms require an initial setting of parameters before starting.

On the other hand, conventional ant colony–based clustering methods tend to leave certain data items unclustered or isolated. Since this thesis focuses on data clustering

(or student team formation), one of the key challenges is ensuring a complete partition of the dataset without any remaining outliers.

To tackle these issues, we propose a structured research approach comprising of four interrelated contributions. The first contribution establishes the theoretical and methodological basis through systematic literature review, which is necessary to move forward. The second contribution builds directly upon this foundation to select relevant attributes, enabling us to explore the selected educational datasets and their features in greater depth. The third contribution presents a comparative study of ant colony clustering parameters' effects and their influence on small and large educational datasets. The final contribution leverages the findings from previous stages to propose a hybrid bio-inspired Ant Colony-based clustering algorithm integrated with the deterministic K-means algorithm (KM-AC) to form synergistic student groups based on academic and social attributes, thus offering an optimized solution to the research clustering problem outlined earlier.

Each of these contributions is presented in a dedicated chapter: First, we present the different aspects of collaborative learning and identify the current state-of-the-art research on student group formation in learning environments Abid et al. (2016). Second, we analyze the impact of feature selection techniques on the classification task to identify relevant features that help improve the performance and effectiveness of our proposed approach Abid et al. (2017). Then, we study the parameters' influence, more precisely on the parameter α, which is responsible for adjusting similarity between objects and its effect on small and large educational datasets Abid et al. (2023). Finally, we define our proposed hybrid bio-inspired Ant Colony clustering approach and analyze the influence of dataset size and group composition on clustering performance. In addition, we also analyze how dataset size affects algorithm performance Abid et al. (2025).

Keywords

Machine Learning, Ant Colony Clustering (ACC), K-means, Student Grouping Problem, Educational Datasets, Parameters Sensitivity.

List of publications

- **Abid A.**, Kallel I. and Ben Ayed M., "Teamwork construction in E-learning system: A systematic literature review," 2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET), Istanbul, Turkey, September 8-10, 2016. IEEE, 2016, pp. 1–7. isbn: 978-1-5090-0778-3.
  DOI: https://doi.org/10.1109/ITHET.2016.7760756.
  Rank: C

- **Abid A.**, Kallel I., Blanco I.J., Benayed M. (2018) Selecting Relevant Educational Attributes for Predicting Students' Academic Performance. In: Abraham A., Muhuri P., Muda A., Gandhi N. (eds) Intelligent Systems Design and Applications. ISDA 2017. Advances in Intelligent Systems and Computing, (AISC) vol 736. Springer, Cham., pp 650-660.
  DOI: https://doi.org/10.1007/978-3-319-76348-4
  Rank: C

- **Abid A.**, Kallel I, Sanchez-Medina J., and Ben Ayed M, (2023) "Parameters Sensitivity Analysis of Ant Colony based Clustering: Application for Student Grouping in Collaborative Learning Environment" in IEEE Access
  Quartiles: Q1 Impact factor:3.476
  DOI: https://doi.org/10.1109/ACCESS.2023.3279723
  SJR: https://www.scimagojr.com/journalsearch.php?q=21100374601tip=sidclean=0

- **Abid A.**, Somai M., Kammoun H. M. and Kallel I. (2024). "The NAJEH Effect: How ChatGPT is Shaping the Future of Higher Education". 21st International Conference on Information Technology Based Higher Education and Training (ITHET), Paris, France, pp. 1-8,
  DOI: https://doi.org/10.1109/ITHET61869.2024.10837661

- **Abid A.**, Kallel I, Sanchez-Medina J., and Ben Ayed M, (2026) "Improving Ant Clustering Algorithms through Supervised Method: Application in Student Grouping with Various Real Datasets", Evolutionary Intelligence. ISSN: 18645909, 18645917 (Submitted)
  Quartiles: Q2 Impact factor : 0.61
  SJR: https://www.scimagojr.com/journalsearch.phpq=14500154734tip=sid

Chapter 1

Student Group Formation in Collaborative Learning Environment

Contents

1.1 Introduction

Forming appropriate groups is one of the fundamental pillars of collaborative learning. As stated in Abid et al. (2016), this issue has inspired many researchers in the field of education to apply various algorithms in order to address the problem of creating learner groups. Some researchers rely on computational approaches Chen and Li (2024), while others draw on heuristic ideas inspired by entomological studies of living organisms in nature Kiran et al. (2022).

Among various collaborative systems, ant-based clustering techniques Priyadarshi and Kumar (2025), Figueiredo et al. (2019) have been successful in solving clustering problems, as demonstrated in Baltierra et al. (2022), Veloz et al. (2019), Lewicki and Pancerz (2020),and Zang et al. (2021). These techniques draw inspiration from the social behavior of real ants, where each ant belongs to a specific category.

In recent years, ant-based clustering techniques have garnered significant attention from the research community due to their performance in exploratory data analysis across many fields, including collaborative robotics as in Kallel et al. (2008), Chatty et al. (2011), Chatty et al. (2012), and Chatty et al. (2013). However, further investigation is needed to address issues related to convergence.

In this chapter, we define the collaborative learning aspects as well as the impact of student grouping and explores common clustering algorithms to support group formation.

1.2 From Learning to Collaborative Learning Systems

Learning can be conceptualized as a process of adaptive transformation arising from the continuous interaction between the individual and the surrounding environment. In fact, learning environment refers to the interaction of three factors: diverse physical locations, contexts of learning and cultures for what/how is the learning process Lage et al. (2000); Kille et al. (2015). It is often used as a preferred and an accurate alternative to the classroom that cope the traditional and limited notion of learning: a simply room with rows and a chalkboard. However, it has enlarged and reaches out beyond the classroom. It generally refers to a more exhaustive definition since learning environment is actually an educational approach that possesses the knowledge experienced by the student, how learners interact/treat one another and it may involve the class, the culture of the school, the population it serves, etc. Learning environments are highly diverse in use, learning styles, organization, and educational institution. Thus, students learn in many different

ways and in very different contexts. Since their common objective is to strive to learn, so a learning environment has to be optimized. Although, there is no single optimum-learning environment, there is an infinite number of possible learning environments. In recent years, it has become widely accepted that students learn more effectively when they collaborate with others Pattanpichet et al. (2011), which explain the increasing use of collaborative learning in educational settings and research works Havard et al. (2008). In fact, grouping learners in teamwork allows them to collaborate, to discuss and exchange their ideas in order to solve problems, which contribute at improving the learning performance of the group as well as of the individuals Huang et al. (2012)Wang and Hwang (2012). However, grouping learners is a challenging task. Hence, to create a smooth collaborative learning, students need to feel safe, comfortable and assigned to the suitable team taking into account their educational background as well as their social attributes. Zheng et al. (2018); Taniguchi et al. (2018).

1.3 Benefits of Collaborative Group Work

Collaboration is a philosophy of interaction and personal lifestyle where individuals are responsible for their actions, including learning and respect the abilities and contributions of their peers. There is a sharing of authority and acceptance of responsibility among group members for the groups' actions. Collaborative group work will bring a supportive and safe learning environment for learners to be more active and patient. They can learn to work cooperatively and they absolutely respect each other, their opinions and liabilities. In group work, they can take this opportunity to apply their technical skills, knowledge as well as their experiences. Especially they can learn and teach each other. So they can learn to work with each other instead of relying on their teachers. They can also realize that getting knowledge is possible without teacher. In group work the role of teacher is the planner, supervisor and organizer. Learners can understand each other and they become social not individualist. Their self-esteem improves and learners motivation increases which intended to improve the quality of learning. The underlying premise of collaborative learning is based upon consensus building through cooperation by group members, in contrast to competition in which individuals best other group membersLaal and Ghodsi (2012). Johnsons' survey Johnson and Johnson (2009) of educational research demonstrates cooperation, in comparison with competitive and individualistic efforts, results in:

- *Higher achievement and greater productivity.*

- *More caring, supportive, and committed relationships.*
- *Greater psychological health, social competence and self esteem.*

1.4 Impact of student group formation in collaborative learning environment

Collaborative learning is a pedagogical strategy that emphasizes active participation, shared responsibilities and mutual support among students . As shown in Figure 1.1, collaborative learning environment attract researchers' attention for the past decade due to the new avenues that Collaborative learning opens to learners and its positive impact on long-term retention and application of knowledge. However, When considering the impact of collaborative learning activities, it is important to take into account how groups has been designed. Therefore, this section discuss the impact of grouping students in collaborative learning environment in several aspects. The study of Alharbi et al. Alharbi et al. (2022) aimed to investigate the influence of an E-collaborative learning environment on the development of critical thinking (higher-order thinking) skills on female students of kinder-garden department who were randomly assigned to two equal groups. Results show a significant and positive effect on various aspects of critical thinking skills of one of the two groups. Furthermore, the study highlights the positive influence of E-collaborative learning on skills such as formulating crucial research questions, identifying relevant variables and theories, describing experimental designs, and enhancing decision-making and problem-solving abilities. Authors in Sung and Hwang (2013), integrate a grid-based Mindtool into a collaborative game-based learning environment in order to allow elementary school students to share and organize what they have learned during the game-playing process. Experimental results confirm that collaborative educational game improve the students' performance in terms of their learning attitudes, learning motivation, self-efficacy and learning achievements. Similarly, the research work presented in Kim (2021), assess the influence of collaborative learning within a virtual environment on high school students for a Korean history class who were randomly assigned to either the "collaborative group" or the "individual group." The findings indicated that students participating in collaborative groups demonstrated significantly higher academic achievement and satisfaction compared to those in individual groups. In the same way, the objective of the research study published in Cheng et al. (2021) is to understand the collaborative learning process when students participated in the flipped learning. The interviews conducted for the participated students

suggest that assisting students to find their corresponding groups, influenced team members' engagement, discussion atmosphere, and efficiency. In addition, it also enhances students' innovative ability, empathy, and promote mutual learning.

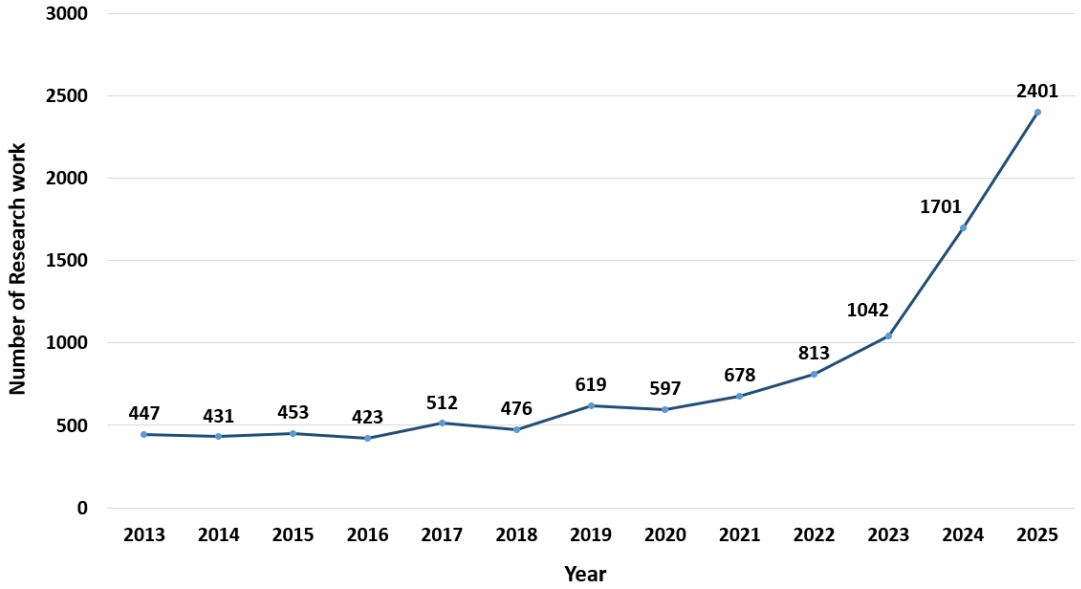


Figure 1.1: Published papers about collaborative learning environment, since 2013. Source: Data extracted from Scopus on Dec.2025

1.5 Common machine learning algorithms used to support group formation in collaborative learning environments

In education settings, the role of machine learning is growing rapidly especially in forming groups due to its positive effect on student performance.

Many researchers are investigating to handle the students grouping problem using machine learning algorithms. In addition, in the field of machine learning, the effectiveness of supervised learning is contingent upon the specific context and techniques employed. Various strategies such as online learning, incremental learning, transfer learning, and ensemble methods contribute to the versatility and adaptability of supervised models. Conversely, real-time adaptability, a crucial attribute in machine learning models, refers to their capability to seamlessly update and adjust to incoming data in real-time. While real-time adaptability is often associated with supervised learning, unsupervised learning can also achieve this feat, albeit through different methodologies. An exemplary approach in unsupervised learning is online clustering, a method well

suited for continuous adaptation. Through continuous updates to clusters, or groups, as new data, or student, arrives, online clustering facilitates real-time adaptability in unsupervised learning scenarios. Table 3 summarize well known machine learning methods according to three taxonomies presented as follows:

Table 1.1: Summary of Known Machine Learning Methods used in Collaborative Learning

Taxonomy	**Machine Learning Method**	**Learning Model**
Computational methods	PC & CS Bourkoukou et al. (2019) Magic Square Peng et al. (2020) K-means Joseph et al. (2017a), EL MEZOUARY et al. (2019), Pang et al. (2014) Decision Tree Gyimah and Dake (2019)	Supervised
Bio-Inspired approaches	GA Chen and Kuo (2019), Zheng et al. (2018), Sukstrienwong (2017),Pawar et al. (2024) Revelo-Sánchez et al. (2021) L & F Boryczka (2009) ACA Boryczka (2009) ACAM Abid et al. (2023), Boryczka (2009) Improved ACA Abid et al. (2023), Gao (2016)	Unsupervised
Hybrid approaches	ACO&LS Badoni et al. (2023)	Optimized unsupervised
	PSO&GA Zheng et al. (2016)	Hybrid Unsupervised
	Substractive&Fuzzy C-Means Yadav (2020)	Hybrid Supervised Fuzzy
	Random Oversampling&Adaboost Hassan et al. (2020)	Hybrid Supervised

1.5.1 Computational Methods

In Peng et al. (2020), authors focused on building heterogeneous groups according to the academic performance levels of students. Their proposed method based on magic square approach was experimented on different datasets sizes. Its findings proved that their proposed approach is a self-grouping methods for generating better heterogeneous groups and can generate several grouping results where users can select the preferred solution. In contrast, authors in Joseph et al. (2017b) proposed an approach based on K-means algorithm where preferences of learners were taken into account in order to form heterogeneous learners groups. Similarly in Pang et al. (2014), authors proposed a technique that employ balanced K-means algorithm to cluster students into homogeneous groups. As stated in Li and Luo (2014), authors proposed an automatic students grouping methods by building a profile for each student based on their knowledge levels, then the clusters are created using network community (NC). Their proposed method was evaluated by comparing the learning effectiveness obtained by the clustered groups

against manually formed groups. While, in the study presented in Joseph et al. (2017a) and EL MEZOUARY et al. (2019), k-means were used for forming groups according to learners' preferences. On one hand, the objective presented in Joseph et al. (2017a) is supporting learners' preferences and enhancing learners' satisfaction in collaborative learning contexts. After applying the k-means algorithm for creating groups, the authors applied an algorithm to generate balanced groups, so that each group can benefit from a good academic performance and good communicative skills. On the other hand, authors in EL MEZOUARY et al. (2019) proposed a new k-means-based approach applied on MOOC environment. The obtained results justify its efficiency to enhance group formation in collaborative learning environment. Nonetheless, the k-means clustering algorithms are incapable of grouping big data with high dimensionality. Moreover, they seek for more improvements in term of their capability to determine the initial number of clusters. Furthermore, other computational methods were proposed in Bourkoukou et al. (2019); Peng et al. (2020); Gyimah and Dake (2019). Where in Bourkoukou et al. (2019), the authors proposed a new approach for automatically grouping learners based on their profiles' heterogeneity. They used the Pearson correlation coefficient and cosine similarity to compute similarity between learners and grouping them according to five similarity levels, namely: Dissimilar, Weak similar, Average similar, Strong similar and Similar. Also in Peng et al. (2020), the authors focused on building heterogeneous groups according to the academic performance levels of students. Their proposed method called MASA and based on magic square approach. It is applied on several datasets with different sizes and they ensured that it's an adaptive and user-friendly method.

Similarly, authors in Gyimah and Dake (2019) defined a new classifier based on decision tree algorithm. This classifier showed high accurate prediction of students' groups when it was evaluating using only one small dataset.

1.5.2 Bio-Inspired Approaches

Recently, algorithms inspired by nature for clustering are in continuous progression Chniter et al. (2018). Accordingly, different bio-inspired algorithms have been applied in several research studies. Stating as examples in Sukstrienwong (2017), Chen and Kuo (2019), Zheng et al. (2018), Pawar et al. (2024) and Revelo-Sánchez et al. (2021) genetic approach (GA) was the main method focused on. In Chen and Kuo (2019), authors proposed a new algorithm called GAGFS-PF. Its aim was to form mixed groups of students by considering some of their homogeneous attributes and other heterogeneous. Their experimental results showed that GAGFS-PF outperformed the other traditional group

formation methods. As well as in the study of Sukstrienwong (2017), authors presented a new system based on genetic algorithm enables teacher to form heterogeneous groups. Their grouping system showed a high level of performance and it is better than the self-selecting traditional method. However, this approach lacks more stability and scalability investigations. Similarly, authors in Zheng et al. (2018) applied genetic approach to propose an improved version in order to construct heterogeneous groups of students. Their experimental results emphasised that this approach is efficient and effective only when it is applied with small datasets. Also, in Pawar et al. (2024), authors proposed a student group formation approach, emphasizing intra-group and inter-group knowledge transfer, supported by a mathematical model optimized using real-coded genetic algorithm (RCGA). Their results show a significant improvement in the fitness function, with an increase in the average fitness value highlighting the robustness of the proposed model. In Revelo-Sánchez et al. (2021), authors proposed an approach based on Genetic Algorithm to group engineering students into homogeneous groups based on their personality traits. Experiments show that their proposed technique gave better academic results when applied on students' personality traits than when applied on by students' preference.

Recently various bio-inspired algorithms such as ant-based clustering represents an interesting approach to perform dynamic grouping Boryczka (2009); Gao (2016). A new version of clustering algorithm were proposed in Boryczka (2009) called ant clustering algorithm modified version (ACAM) inspired from Lumer and Faieta algorithm (L&F) Lumer and Faieta (1994). Then, another advanced method is proposed in Gao (2016) called improved ant clustering algorithm (Improved ACA) to solve grouping problems. All of them evaluated their proposal using classical datasets from UCI machine learning repository and presented high accuracy of clustering.

Furthermore, authors in Abid et al. (2023) investigate more in Ant Colony based clustering algorithm performance when applied in educational settings for clustering groups of learners as illustrated in Figure **??**. Their findings prove that ACAM and Improved ACA algorithms clustering performance depends on data size. In addition, the deep analysis of these algorithms' parameters attest that Ant Colony algorithms performance intensely depend on α parameter.

1.5.3 Hybrid Approaches

In recent years, hybrid clustering methods are inspiring more and more researchers especially in the last decade. As reported in Figueiredo et al. (2019), the use of hybrid swarm-intelligence clustering approaches with the standard clustering algorithms lead

to high clustering performance achievements. For example, authors in Badoni et al. (2023) proposed a new way of hybridizing of Ant Colony Optimization (ACO) algorithm with the local search (LS) in order to create mutually exclusive groups of students. Their main goal is to assign each student into exactly one group.
Similarly, authors in Zheng et al. (2016) proposed a group formation method entitled Hybrid PSO-GA that aims at integrating Particle swarm optimization (PSO) algorithm with the crossover and mutation operators of Genetic Algorithms. The two hybrid methods presented competitive and high quality achievement solutions. One more study in Yadav (2020), proposed a hybrid clustering approach for evaluating the assessment of students based on their intelligence level in the educational environment. This hybrid method is based on integrating techniques of subtractive and Fuzzy C-Means clustering methods and it showed better classification outputs when compared to other methods. One more similar study in Hassan et al. (2020) presented a hybrid technique of Random Oversampling with AdaBoost in order to improve the students' performance prediction model. This work showed that composite techniques are vital for improving the accuracy of predictive models and single classifiers. However, it still needs some other combinations of hybrid techniques to deal with the multi-class classification problem.

1.6 Overview of Ant-based Data Clustering: Presentation of well-known Ant Clustering algorithms

In this section, we present an overview on data clustering techniques and the contributions of Ant Colony Clustering (ACC) systems in solving clustering problems. Then, we present the ant-based data clustering algorithms that we depend on by after for grouping student in collaborative learning environment.

1.6.1 Nest Organization Algorithms for Data Clustering

Many researchers returned to some entomologists like Anari et al. (2018) and Boryczka (2009) who study the ant colonies to recognize the behavior of real ants in forming cemeteries. The next parts define the algorithms arise from this behavior and how they achieve the goal of clustering data.
Hence, the activities of ants' cemeteries construction is summarized in the following activities that illustrated in figure 1.2. At the beginning, the dead ants are scattered in the nest. Hence, in order to clean the nest from corpses and cluster them, each ant start by walking randomly around the space. If it hits one of corpses, the ant have two

choices: to pick up the corpse if it is not in a dense location of dead ants, then to move it and drop it where they are surrounded by other similar dead ones. The second choice is to let it if it is already surrounded by other similar ones. The ant distinguishes its living nest-mates from dead ones through their specific corpses odor. The next parts discovers the algorithms arise from this behavior and how they achieve the goal of clustering data.

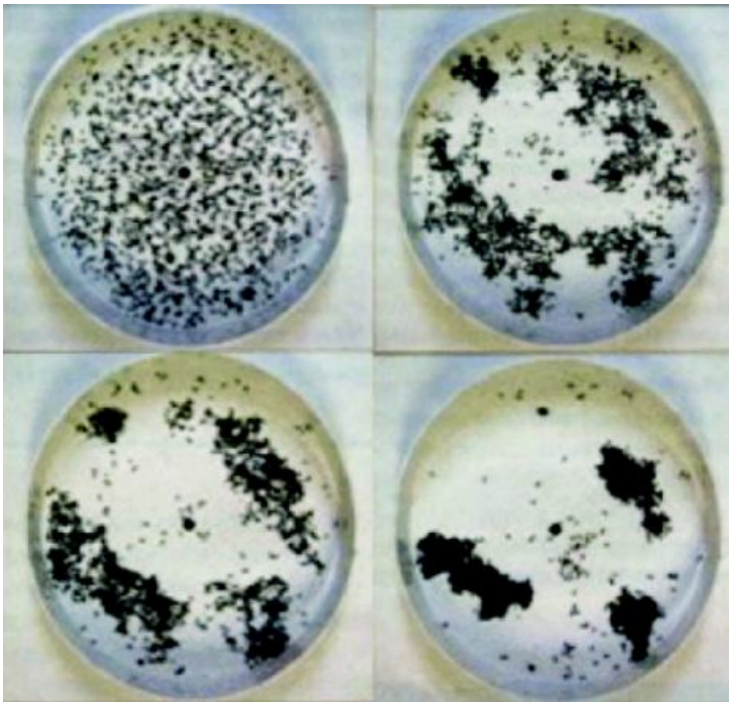


Figure 1.2: the real ant clusters Bonabeau et al. (1999)

1.6.2 Presentation of well-known Ant Clustering algorithms

According to Dorigo et al. (2006), swarm intelligence is a relatively new approach taking inspiration from the social collective behaviors of insects and animals. Particularly, ants behaviors gave rise to many methods, among which the most known is the ant colony clustering algorithms, for solving grouping data problems. In the current study, we implement, discuss and extend four well known ant clustering algorithms Boryczka (2009): First, the Lumer and Faieta algorithm (L&F) that can find groups without any previous knowledge about the number of groups; also, it project data onto lower dimensional space: 2D. Second, the ant based clustering algorithm (ACA) that introduces new ideas and modifications in LandF's algorithm to improve the convergence. Then, the Ant Clustering Algorithm Modified version (ACAM) which proves a high quality of performance comparing to the ACA. Finally, the Improved ACA algorithm with better accuracy, efficiency and high level of speed.

L&F algorithm

L&F algorithm proposed by Lumer and Faieta (1994) operates on a fixed regular low-dimensional grid where ants are generated in a simulation environment. Its main idea is to define the measure of similarity and dissimilarity among the different objects/items, then to cluster them into well defined groups. This algorithm contains three major phases which are: the initialization phase, the simulation phase and the cluster phase which are illustrated as following:

1^{st} **phase:** called the initialization phase where the ants are randomly performed on the grid, as well as for items o_i, they are randomly laid on the grid.

2^{nd} **phase:** is the activity simulation phase which is evolved in a discrete time *t* steps as detailed in the following algorithm. Each ant is randomly selected and moved along the grid; if there is a pattern on its current location, it can pick it up, otherwise if the object is already carried, the ant can drop it.
The probabilities of picking up p_{pick} or deposit $p_{deposit}$ a pattern depends on the distance in feature space between the pattern and its neighbors d(i,j) measured as in equations 1.1 and 1.2.

$$p_{pick} = \left(\frac{\gamma_{pick}}{\gamma_{pick} + f}\right)^2 \tag{1.1}$$

and

$$p_{deposit} = \begin{cases} 2\, f(o_i), & when\ f(o_i) < \gamma_{deposit} \\ 1, & when\ f(o_i) \geqslant \gamma_{deposit} \end{cases} \tag{1.2}$$

with f is the perceived fraction of local density of neighboring sites occupied by data points of the same type, γ_{pick} and $\gamma_{deposit}$ are two threshold constants.
Generally, the probability of picking up an object increased if it is surrounded by dissimilar data, or when there is no data in its neighborhood.
The local density with respect to object o_i is given by equation 1.3

$$f(o_i) = \begin{cases} \frac{1}{\sigma^2} \sum_{o_j \in neigh(s*s)(r)} \left(1 - \frac{d(i,j)}{\alpha}\right) & , if\ f \geqslant 0 \\ 0 & , otherwise \end{cases} \tag{1.3}$$

Where α is the scaling dissimilarity parameters that allows the determination when two items should or should not be located next to each other.
The σ represents the neighbor size, $\sigma \in$ [9,25]. And *r* is the ant located side and *s* is the neighborhood scaling parameters.

3^{rd} **phase:** called the clustering phase of objects occurs in order to determine the boundaries between the groups.

Algorithm 1 L&F Algorithm

```
Randomly place o_i on the grid
Place ant at randomly selected site
for all ants do
    for t=1 to t_max do
        if (ant is not carrying a load) & (site is occupied by object o_i) then
            Compute equation 1.3 and 1.2
            Draw a random real number r ∈ [0, 1]
            if α ≤ p_pick(o_i) then
                Pick up o_i
            end if
            if (o_i carried by an ant) and (site empty) then
                Compute equations 1.3 and 1.1
                Draw a random real number r ∈ [0, 1]
                if α ≤ p_deposit(o_i) then
                    Deposit o_i
                end if
            end if
            Move to randomly selected neighboring site not occupied by other ant.


            Output clusters.
```

ACA and ACAM algorithms

The ant based clustering algorithms are first proposed idea by Deneubourg et al. (1992). In accordance with this, a number of modifications have been applied in various advanced studies that can improve the quality of the clustering. Hence, in this part, we present the modified versions of L&F for data clustering problem proposed in Boryczka (2009).

The Phases

The Ant Clustering Algorithm (ACA) and the The Ant Clustering Algorithm Modified version (ACAM) exploit the short-term memory of each ant. It is the number *n* of items that come across during the last time *t*. Each artificial ant uses its memory depending on some basics:

If an ant is located at a specific cell on the grid, and carrying an object o_i, it employs its own memory to proceed to all remembered placements, one after the other. Each of them is evaluated using the neighborhood/ density function $f^*(o_i)$ for finding the fitting site to deposit the current carried object. These proposals have also three main phases:

1^{st}**phase**: the initialization phase, the items are randomly dispersed on the grid. Thereafter, for each ant start by selecting the object o_i at a random way, then pick it and

move it to another empty grid location.

2^{nd}**phase**: the simulation phase is evolved too in a discrete time *t*, the ants are randomly selected and moved to other new locations. Then, the ants carried the items and measure the similarity between them and the probability of deposit an object. They are computed according to equation 1.4:

$$f^*(o_i) = \begin{cases} \frac{1}{\sigma^2} \sum_j \left(1 - \frac{d(i,j)}{\alpha}\right) & ,if\ f^* \geqslant 0\ and\ \sum_j (1 - \frac{d(i,j)}{\alpha}) \geqslant 0 \\ 0 & ,otherwise \end{cases} \tag{1.4}$$

with σ is the neighbor size. For dropping decision, a threshold is expressed by equation 1.5 is used.

$$p^*_{deposit}(i) = \begin{cases} 1, & if\ f^*(i) \geqslant 1 \\ \frac{1}{f^*(i)^4}, & else \end{cases} \tag{1.5}$$

Subsequently, if the operation of dropping the item is done; the ant continue selecting the object o_i, pick up it, move it and compare it with others surrounded ones or move it to another empty location until they find the suitable sites of data. The picking probability is calculated by equation 1.6.

$$p^*_{pick}(i) = \begin{cases} 1, & if\ f^*(i) > 1 \\ \frac{1}{f^*(i)^2}, & else \end{cases} \tag{1.6}$$

3^{rd} **phase:** called the cluster phase in which the defined clusters become to well appeared.

As the same steps, the authors in Boryczka (2009) proposed another modified version of ACA called ACAM based on the following new modification using a new neighborhood scaling parameter $\frac{S_0^2}{S^2}$ depends on an adaptive perception range *S*.

The density /neighborhood function in the new version ACAM is computed by equation 1.7:

$$f^*(o_i) = \begin{cases} \frac{S_0^2}{S^2} \sum_j \left(1 - \frac{d(i,j)}{\alpha}\right) & ,\forall o_j (1 - \frac{d(i,j)}{\alpha}) > 0 \\ 0 & ,otherwise \end{cases} \tag{1.7}$$

where S^2 represents the perception range of each ant. Thence, $\frac{S_0^2}{S^2}$ is the relation between the initial size of perception S_0 and the current size of perception S, it defines the new quarter scaling parameter.

Consequently, the ACAM solution is proposed in order to overcome the difficulty to

determine and distinguish the differences between clusters of different sizes. Because on the previous studies of L&F and ACA, they presented a stable values of neighborhood scaling parameter $\frac{1}{\sigma}$ which may lead to inappropriate results of clusters.

Algorithm 2 ACA Algorithm

```
the items o_i are randomly scattered on the grid
for each ant do
    Randomly select the items o_i
    Pick up o_i
    Place the ant at random selected empty grid
end for
for t = 1 to t_max do
    Randomly selected an ant
    Move it to another new location
    Ant carried out the object o_i
    Calculate f*(o_i) and p*_deposit(o_i)
    if Deposit = true then
        while pick = False do
            Randomly select items o_i
            Calculate f*(o_i) and p*_pick(o_i)
            Pick-up o_i
        end while
    end if
end for
    =0
```

Improved ACA Algorithm

Based on the ACA solution, the author Gao Gao (2016) proposed a new Improved ACA and Abstraction Ant Clustering Algorithm (AACA) presented more efficiency and accuracy than the previous proposals. So, we choose to define the Improved ACA proposal because it is required less number of parameters than AACA and without applying the combination mechanism. That can slow the average time of algorithm.

The process behind this algorithm is similar to the previous three mentioned algorithms with some updates in the similarity function and probabilities measures, it is described as follow:

1^{st}**Stage:** is the initialization phase.

- **(1)** Initialize all of these parameters: the number of ants *n*, the number of iterations *m*, the local region length *s*, the α that measure the similarity between each two data items and the slope constant c that can speed up the convergence of the algorithm and the maximum speed v_{max}.
- **(2)** Initialize a random pair of coordinates (x,y) to get a random scattered data items on the grid.

- **(3)** the ants start moving along the grid and each one chooses an object randomly.

2^{nd}**Stage:** At a given maximum number of iterations *m*, each ant measures the average similarity between items. this function is defined by equation 1.8.

$$F^*(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j \in neigh(s*s)(r)} [\, 1 - \frac{d(o_i,o_j)}{\alpha(1+(v-1)/v_{max})} \,] & , \\ 0 & , otherwise \end{cases} \tag{1.8}$$

With s^2 equals to the total number of sites in the local area of ant. Each ant characterized by a speed $v \in [1, v_{max}]$.

The process behind the picking up and dropping probabilities in this algorithm is as follow:

- If the ant's mouth is empty, the picking up probability P_{pick} is computed. If it is greater than a random number probability and an object is not simultaneously picked up by another ant, the ant picks up this object, marks itself as its mouth is not empty, and moves this object to a new position; otherwise, the ant does not pick up this object and randomly selects another object Gao (2016). The function of picking up probability is calculated as by equation 1.9.

$$P_{pick} = 1 - sigmoid(f(o_i)) \tag{1.9}$$

- If the ant's mouth is not empty, the dropping probability P_{drop} is computed. If P_{drop} is greater than a random probability, the ant drops the object, marks itself as unloaded, and randomly selects a new object; otherwise, the ant continues moving the object to a new position Gao (2016). The function of deposit probability is calculated by equation 1.10

$$P_{drop} = sigmoid(f(o_i)) \tag{1.10}$$

the sigmoid function is expressed by equation 1.11.

$$sigmoid(x) = \frac{1 - e^{-cx}}{1 + e^{-cx}} \tag{1.11}$$

3^{rd}**Stage:** is the output of clusters. If an object is isolated, it is labeled as an outlier. otherwise, give this object a cluster labeling number and recursively label the same number to those items that are neighbors of this object within the local region as

described Gao Gao (2016).

Algorithm 3 Improved ACA Algorithm

Initialize the parameters: n, m, s, $alpha$, c, v_{max}
Initialize a random pair of coordinates (x,y) to get a random scattered data items on the grid.
for each ant **do**
    Randomly select the items o_i
    Pick up o_i
end for

for i = 1 to M_{max} **do**
    for j=1 to N **do**
        Compute $F^*(o_i)$
        if (ant mouth is empty) & (site is occupied by object o_i) **then**
            Compute P_{pick}
            if $P_{pick} > \alpha \in [0, 1]$ **then**
                pick up o_i
                ant mouth is not empty
                ant moves o_i to a new location.
            end if
        end if
        Compute $F^*(o_i)$
        if (ant mouth is not empty) **then**
            Compute P_{drop}
            if $P_{drop} > \alpha \in [0, 1]$ **then**
                drop o_i
                ant mouth is empty
                ant moves to a new location
            end if
        end if
        Move to randomly selected neighboring site not occupied.
    end for
end for

Output clusters

=0

1.6.3 Assessment and Discussion of the Selected Algorithms

Considering the previous descriptions of algorithms, we can extract that these algorithms differ in the following points:

- In **L&F** each ant is characterized by an area called a neighbor size. When it selects an item when moving randomly and coming in crowds with similar surrounding items, the ant choose to drop the selected object.
- In **ACA and ACAM** each ant is characterized by a memory to reduce the random trends. When a new is picked up by an ant, a comparison process is made with the items in memory. So, it automatically moves towards the location of the memorized items most similar to the picked one.

- In **Improved ACA**, the algorithm keeps the same functionality as L&F but each ant characterized by a specific speed v distributed randomly [v,V_{max}].

The most common point that characterized these algorithms that they are automatically determining clusters without any prior knowledge about the possible number of clusters as introduced by Kao and Li Kao and Li (2008), Boryczka Boryczka (2009), Gao Gao (2016). However, Ant colony clustering algorithms are categorized as unsupervised learning model. Inspired by the collective behavior of ants, which has been refined through millions of years of evolution, making them inherently robust and effective in solving complex optimization problems. These algorithms offer several advantages such as their efficiency in finding good clustering solutions, especially in large, non-linearly, high-dimensional and complex datasets, etc. Furthermore, these algorithms are not restricted to specific data types or clustering criteria and can be applied to various types of data and clustering objectives. Their adaptability to incorporate domain-specific knowledge as we can mention their application in a large variety of application fields, such as Road Traffic Management Kammoun et al. (2011), electricity-theft detection Yang et al. (2024) and students' academic performance analysis Xu and Kim (2024). In addition, these algorithms are robust to dynamic environments where they can adapt to changes in the dataset or problem domain by continuously updating the clustering solution. Although all the mentioned above advantages, the stochastic nature of ant colony clustering algorithms allows them to escape local optima and explore a diverse range of clustering solutions and they are counted as non deterministic algorithms which mean that some items my not be assigned to a cluster. In our case, since we apply ant clustering algorithms on educational settings, students should not be considered as outliers and they must be assigned to a group of learners.

1.7 Conclusion

Collaborative learning is a pedagogical strategy that fosters active participation, shared responsibilities, and mutual support among students. It encompasses various educational approaches that encourage learners to work in groups to exchange knowledge, resources, and ideas, thereby enhancing both individual and group performance. Effective group formation is a critical factor in maximizing the benefits of collaborative learning. In this chapter, we provide an overview of on the impact of student grouping in collaborative environment environments and explore clustering algorithms that to support group formation. In addition, we also present four well-known heuristic approaches based on ant colony clustering algorithms : L&F, ACA, ACAM, and the Improved ACA. These are

algorithms counted as non-deterministic , algorithm and , a key requirement in educational contexts is that each student be must assigned to a distinct cluster, with ensuring no outliers remaining remain after clustering. Accordingly, subsequent chapters deepen this research to identify solutions ensuring every learner belongs to a specific.

Chapter 2

Selecting Relevant Features in Educational DataSets

Contents

2.1 Introduction

The secret that lies behind the apparition of the emerging interdisciplinary field Educational Data Mining (EDM) is the huge increasing of educational data in institutions. However, an immense amount of data with different formats and from multiple sources may contain a large number of features supposed as not-relevant that could influence the efficiency of learning systems. In this chapter, we carry out a comparative study for evaluating the influence of feature selection techniques which can play an important role in improving the data quality therefore the performance of the clustering algorithms.

2.2 Relevant features selection in educational data

Feature Selection is one of the prominent preprocessing steps in many research area such as pattern recognition, machine learning and data mining communities Mitra et al. (2002) Miller (2002). In fact, these techniques select the relevant features from the original feature set according to an evaluation criterion. It aims to reduce dimensionality, remove not-relevant or redundant data, increase learning accuracy, etc.

2.2.1 Overview on Known Feature Selection Processes

Among several available feature selection techniques, we select to present in this paper four techniques, namely:

Correlation Based Feature Selection (CFS): Hall (1999) is a heuristic for evaluating the merit of features subset. Its main idea is selecting feature subsets including attributes that are highly correlated to the class, yet uncorrelated with each other. In other word, the heuristic handles not-relevant features as they will be poor predictors of the class. In addition, the redundant features will be excluded as they will be highly correlated with one or more of the remaining features. Equation 2.1 formalises the heuristic:

$$Merit_s = \frac{k\overline{r_{ca}}}{\sqrt{k + k(k-1)\overline{r_{aa}}}} \tag{2.1}$$

Where $Merit_s$ models the correlation between the summed attributes and the class variable. s is an attribute subset that has k attributes. $\overline{r_{ca}}$ is the average of correlation between the attributes and the class variable. Finally, $\overline{r_{ca}}$ is the average inter-correlation between attributes.

ReliefF (RF): is an instance-based attribute ranking. It was introduced by Kira and Rendell Kira and Rendell (1992) and later enhanced by Kononenko Kononenko (1994).

This algorithm evaluates the worth of an attribute by ranking it with a value between -1 and 1, more positive weights indicates more the predictively of the attribute. As shown in the pseudo code 0, for each instances m, Relief randomly sampling an instance from the dataset. Then, it locates its nearest neighbor (H_j) from the same and opposite class (M_j). Finally, it updates the relevance scores for each attribute by comparing the weight of the attributes of the nearest neighbors and the sampled instance.

Information Gain (IG): In probability and information theory, IG is a measure of the difference between two probability distributions. This method evaluates an attributes by measuring its information gained with respect to the class. In other word, IG is an amount by which the entropy of the class decreases reflects the additional information about the class provided by the attribute Quinlan (2014).
If A is an attribute and C is the class, the entropy of the class before and after observing the attribute is presented in Equations 2.2 and 2.4.

$$H(C) = -\sum_{c \in C} p(c) log_2 p(c) \tag{2.2}$$

$$H(C \mid A) = -\sum_{a \in A} p(a) \sum_{c \in C} p(c \mid a) log_2 p(c \mid a) \tag{2.3}$$

For each attribute A_i , a score is assigned based on the information gain between itself and the class (See Equation 4).

$$\begin{aligned} IG_i &= H(C) - H(C \mid A_i) \\ &= H(A_i) - H(A_i \mid C) \\ &= H(A_i) + H(C) - H(A_i \mid C) \end{aligned} \tag{2.4}$$

Symmetrical Uncertainty (SU): Press et al. (1996) uses an information theoretic measure to evaluate the rank of an attribute by Equation 2.5. This method is symmetric in nature, $SU(A, C)$ is same as that of $SU(A, C)$, which help at reducing the number of comparisons required especially when A and C are two features.

$$SU(A, C) = 2[\frac{IG(A \mid C}{H(A) + H(C)}] \tag{2.5}$$

Where $IG(A|C)$ is the information gain of feature A, that is an independent attribute and C is the class attribute. $H(A)$ is the entropy of attribute A and $H(C)$ is the entropy of feature/Class C. It's worth to mention that SU is not influenced by multivalued attributes as is the case for IG.

Algorithm 4 A pseudo code of Relief feature selection Kononenko (1994)

Set all Weights W[A] = 0.0
for i=1 to m **do**
  Randomly select an instance R
  Find K nearest hits H_j
  for each class C $\neq$ class (R) **do**
    find K nearest misses $M_j(C)$
  end for
  for A=1 to $\sharp$ attributes **do**
$W[A] = W[A] - \sum_{j=1}^{k} diff(A, R, H_j)/(m \times k) + \sum_{c \neq class(R)} [\frac{P(C)}{1 - P(class(R))} \sum_{j=1}^{k} diff(A, R, M_j(C))/(m \times k)$
  end for
end for=0

2.2.2 Litterature Review of Some Feature Selection Methods

Feature selection process can play an important role in improving the performance of the learning algorithm. Several researchers have performed these techniques before applying classification algorithms in many application domains such as Education, Security, etc.

In Ramaswami and Bhaskaran (2009) the authors carried out a comparative study of six well-known filter feature section algorithms in order to improve the effectiveness of a students' model that could predict their academic performance. They were able to define the best method and reach an optimal dimensionality of the feature subset. The finding of their work show a reduction in computational time and constructional cost in both training and classification phases of the student performance model and finally they deduced that "Information-Gain Attribute evaluation" is the best feature selection technique for their predictive model. Similarly, the main purpose presented in Velmurugan and Anuradha (2016) is to enhance the results of predicting students' performance in the final semester examination. To achieve their goal, the authors analyzed the impact of feature selection techniques on the classification task in order to identify the best one. They compared four feature selection algorithms, and based on the results, they selected "CFS Subset Evaluator" as the best feature selection algorithms.

Authors in Costa et al. (2017) carried out a comparative study on the effectiveness of four feature selection techniques to early predict students likely to fail in introductory programming courses. For this purpose, they applied the four techniques on two different and independent data sources on introductory programming courses. The outcome of this comparative study selected "Information Gain algorithm" since it presented the

best results in both data sources.

In the work presented in Bolón-Canedo et al. (2013), seven filters were applied on 11 synthetic dataset with a not-relevant features, redundancy and interdependency between attributes. The authors review the efficiency of feature selection methods and finally they were able to select "ReliefF algorithms" as the best method. Authors in Noura et al. (2016) proposed a new feature selection technique based on random forest. Their aim was to select the most relevent and non-redundant features. They used the random forest to measure the relevance value attributes and the correlation coefficient to calculate the value of redundancy. The proposed approach has been tested and validated on nine different databases.

As a summary, from the previous works, we may notice that there is not a one common feature selection method which can be accurate for all dataset even for the same domain. For this reason, we select a set of four feature selection techniques in order to identify the best attribute selection algorithm for our predictive students' performance model.

2.3 Feature Selection Methodology in Educational Dataset

The main purpose of this chapter is to analyze the impact of feature selection techniques on the performance of the clustering algorithm. Accordingly, we consider four feature selection techniques, introduced in the previous section, are as follows: Correlation Based Feature Selection (CFS), Information Gain (IG), ReliefF (RF) and Symmetrical Uncertainty (SU). These algorithms are combined with five classifiers in order to evaluate their performance. Therefore, a classification model is built using RandomForest Breiman (2001), REPTree Cohen (1995), LogitBoost Friedman et al. (2000), JRip Cohen (1995) and J48 Quinlan (1993).

The methodology adopted for this comparative study is displayed in Figure 2.1. The first step is transforming the available dataset into data format of the destination data mining system, which in this case from Excel to Attribute-Relation File Format (ARFF)1 format of Weka Smith and Frank (2016). As we suspected that the first and second grades of learners would have a high impact on the final results, we decided to divide the dataset into social and academic data. Then, by applying the feature selection techniques on the transformed data, a set of five features sets are the results of this step. Next, balanced data step is needed in order to deal with one of the important problem that may exist in educational data, which is unbalanced data Márquez-Vera et al. (2013). This problem may occur when one class has a number of instances larger than the one in other

classes. Thus, prediction algorithms may focus on learning from them Gu et al. (2008). After preprocessing data and applying the four feature techniques, the collection of five dataset are now ready to be the input of the classification model. Finally, the collected results are used to analyze the performance of each classification algorithms in order to select the best features.

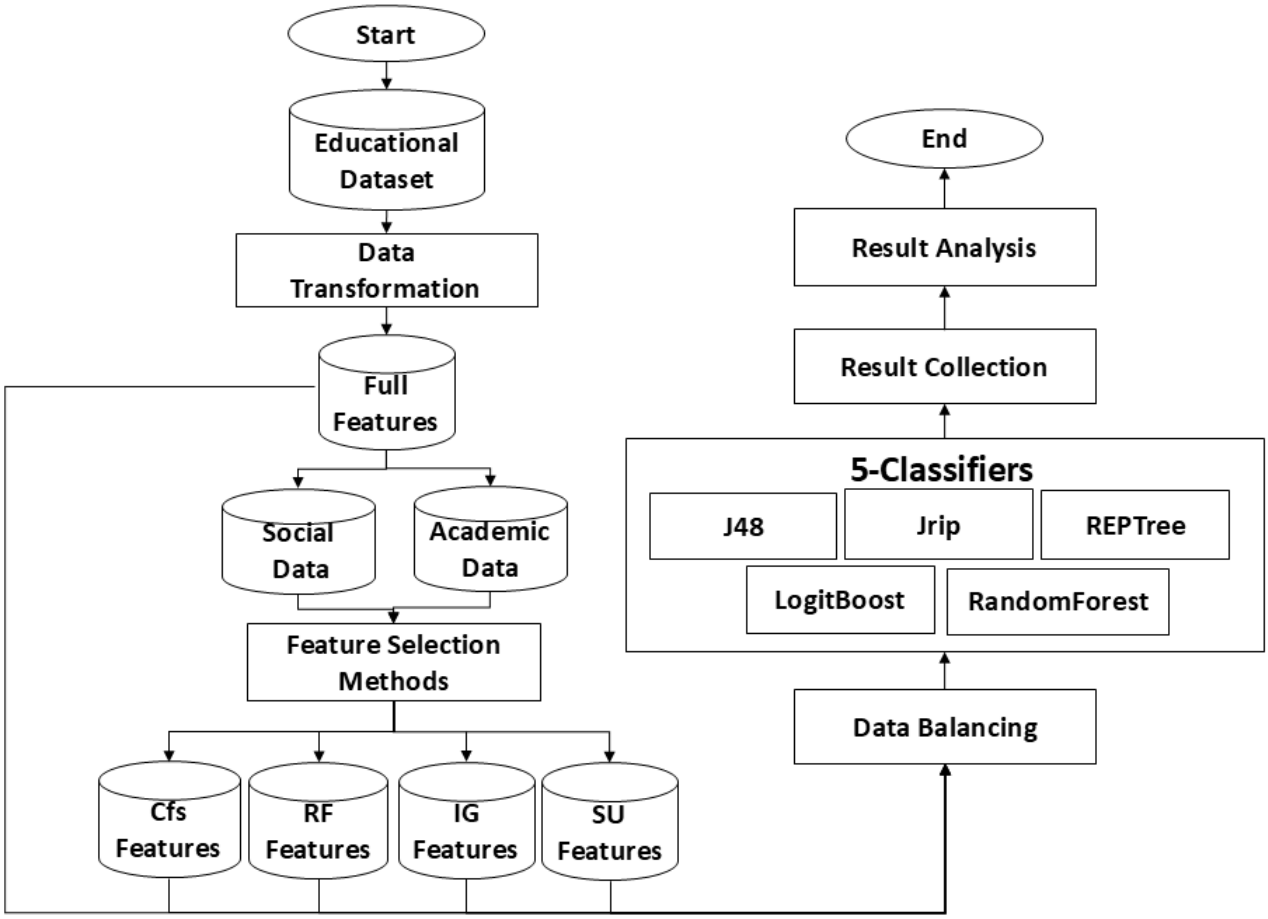


Figure 2.1: Flow chart of the proposed methodology

2.4 Educational Data Preprocessing

In order to experimentally investigate the performance of our proposed metodology, we selected 14 concrete educational dataset with different sizes, which are defined as follows: Two datasets provided by Cortez and Silva (2008b) with 33 attributes (Large Dataset) and datasets collected by Amrieh et al. (2015) Amrieh et al. (2016) with 15 features presenting students' academic and social data (xAPI-datasets)

2.4.1 Educational Data Description

The two datasets provided by Cortez and Silva (2008b) were collected during the 2005-2006 school year from two public schools, from the Alentejo region of Portugal.These datasets were built from school reports, based on paper sheets and including few attributes such as the three period grades and using a questionnaires in order to complement the previous information which was related to several demographic data (e.g. mother's education, family income), social/emotional (e.g. alcohol consumption) and school related (e.g. number of past class failures) variables that were expected to affect student performance. As a result, data were integrated into two datasets containing 33 attributes detailing students' academic and demographic as illustrated in Table 2.1 for a specific core classes: Mathematics dataset (with 395 examples) and Portuguese language dataset (649 records).

For the xAPI-datasets collected by Amrieh et al. (2015) Amrieh et al. (2016) from the multi-agent LMS Kalboard 360, which allows synchronous access to learning resources from any device and involves parents and school management in the learning process. Learner activity is tracked using the Experience API (xAPI), part of the Training and Learning Architecture (TLA), enabling monitoring of learning progress and actions such as reading articles or watching videos, helping providers understand learner activities and experiences. The xAPI-datasets consist of 480 student records with 15 attributes as illustrated in table 2.2. These data were collected through two semesters where students take 12 different learning subjects which are: IT, French, Arabic, Science, English, Biology, Spanish, Chemistry, Geology, Quran, Math History. Figure 2.2 depicts how students are distributed across these 12 subjects.

Since we are grouping learners into teams to study together for the same lecture, we divided this dataset into subsets according to the topic of course.
This results in a total of 12 X-API-datasets in addition to the two Large Dataset mentioned above, Mathematics dataset and Portuguese language for further investigation in this research work.

2.4.2 Data cleaning and transformation

Attributes within a dataset can be correlated, redundant or with no predictive information. Consequently, the data must be preprocessed to select an appropriate subset of attributes and ignore irrelevant and redundant ones. Consequently, a data-preprocessing step is needed in order to prepare it and improve it for further analysis. Data transformation is the process of converting or mapping data from one format into another one

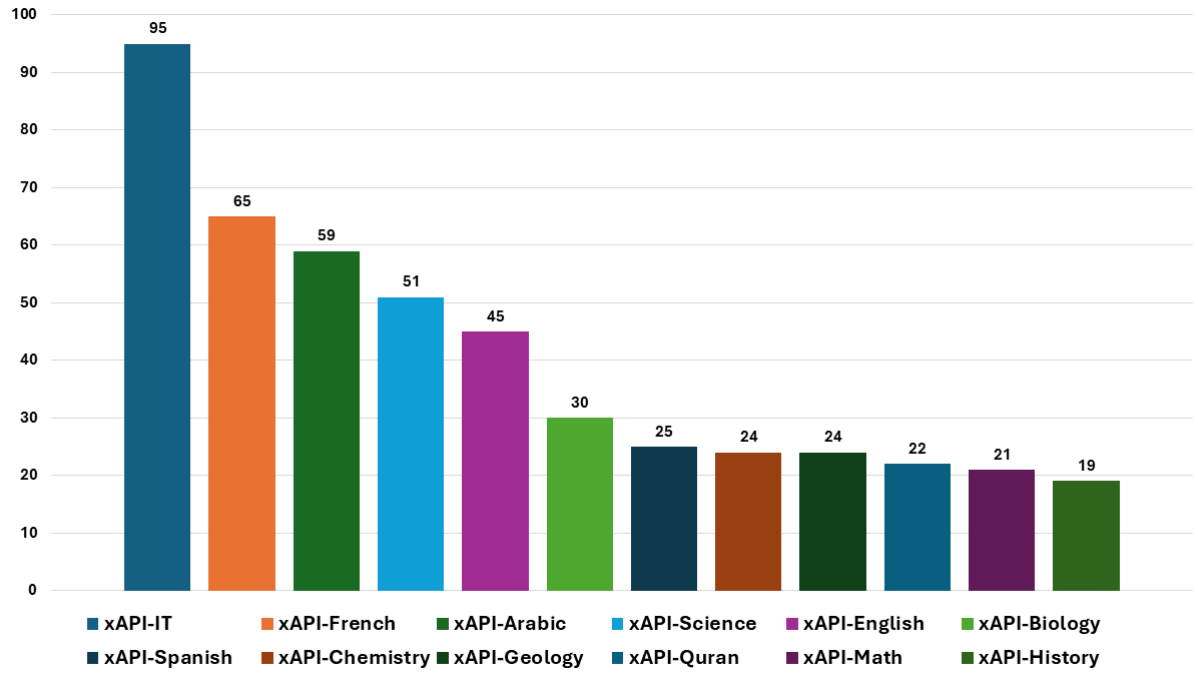


Figure 2.2: 480 student records distributed across 12 learning subjects

to prepare it for further uses. In this section, we apply the data transformation on the available dataset to convert it into data format of the destination data mining system, which in this case from excel to .ARFF format of Weka Smith and Frank (2016). After the preprocessing step, our selected educational data are ready to be used for mining and analysis which will be the subject of the next section.

2.5 Feature Selection: Results and discussion

The performance of a classification model highly depends on the datasets features. Accordingly, in this section, we are going to apply the different steps of the methodology presented above in order to discuss the outcomes and to select the best feature selection technique. The effectiveness of such an algorithm could be presented in different parameters, such as f-Measure, number of features and time taken to build the classification model. As we mentioned previously, in this research study, we are using two datasets that share the same 33 attributes and 12 dataset share 15 features.

Table 2.1: Large dataset related attributes Cortez and Silva (2008b)

Attributes	**Description**
Sex	student's sex (binary: female or male)
Age	student's age (numeric: from 15 to 22)
Address	student's home address type (binary: urban or rural)
Famsize	family size (binary: 3 or > 3)
Pstatus	parent's cohabitation status (binary: living together or apart)
Medu	mother's education (numeric: from 0 to 4a)
Fedu	father's education (numeric: from 0 to 4a)
Mjob	mother's job (nominalb)
Fjob	first period grade (numeric: from 0 to 20)
Guardian	student's guardian (nominal: mother, father or other)
Traveltime	home to school travel time (numeric: 1 – < 15 min., 2 – 15 to 30 min., 3 – 30 min. to 1 hour or 4 – > 1 hour)
Romantic	with a romantic relationship (binary: yes or no)
Famrel	quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
Freetime	free time after school (numeric: from 1 – very low to 5 – very high)
Goout	going out with friends (numeric: from 1 – very low to 5 – very high)
Dalc	weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
Walc	workday alcohol consumption (numeric: from 1 – very low to 5 – very high)
Health	current health status (numeric: from 1 – very bad to 5 – very good)
School	student's school (binary: Gabriel Pereira or Mousinho da Silveira)
Reason	reason to choose this school (nominal: close to home, school reputation,) course preference or other)
Studytime	weekly study time (numeric: 1 – < 2 hours, 2 – 2 to 5 hours, 3 – 5 to 10 hours or 4 – > 10 hours)
Failures	number of past class failures (numeric: n if 1 n < 3, else 4)
Schoolsup	extra educational school support (binary: yes or no)
Famsup	family educational support (binary: yes or no)
Paidclass	extra paid classes (binary: yes or no)
Activities	extra-curricular activities (binary: yes or no)
Nursery	attended nursery school (binary: yes or no)
Higher	wants to take higher education (binary: yes or no)
Internet	Internet access at home (binary: yes or no)
Absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)

2.5.1 Most relevant selected attributes after applying Feature algortihms

By applying the four feature selection techniques (CfsSubsetEval, ReliefAttributeEval, InfoGainAttributeEval, and SymmetricalUncertAttributeEval) on our selected Dataset,

Table 2.2: Small dataset related variables Amrieh et al. (2015) Amrieh et al. (2016)

Attributes	**Description**
gender	The gender of the student (female or male)
NationalITy	Student nationality
PlaceofBirth	Place of birth for the student (Jordan, Kuwait, Lebanon,) Saudi Arabia, Iran, USA)
StageID	Stage student belongs such as (primary, middle and high school levels)
GradeID	Grade student belongs as (G-01, G-02, G-03, G-04, G-05, G-06, G-07, G-08, G-09, G-10, G-11, G-12)
SectionID	Classroom student belongs as (A, B, C)
Semester	Classroom student belongs as (A, B, C)
Relation	Parent responsible for student (Father, Mother)
raisedhands	number of time students raised their hands
VisITedResources	number of time students visited resources
AnnouncementsView	number of time students viewed announcements
Discussion	number of time students participated in discussion
ParentAnsweringSurvey	Parent is answering the surveys that provided from school or not
ParentschoolSatisfaction	This feature obtains the Degree of parent satisfaction from school as follow (Good, Bad)
StudentAbsenceDays	Student absence days (Above-7, Under-7)

we collect four sets (Cfs features, Rf feature, IG features, SU feature) where each of them is the outcome of each techniques. Table 2.3 (respectively Table 2.4), shows the final selected feature for each set as well as the number of the selected attributes of Mathematic course dataset (respectively Portuguese language course dataset and Tables 2.5 (respectively Table 2.6) illustrate results for xAPI-Chemistry Dataset (respectively xAPI-English Dataset) while the other 10 xAPI-Edu dataset results are presented in Tables 1 of Appendix .1. According to mentioned Tables, each feature selection technique gives different results than each other, which make the decision of selecting one of them as the main feature selection technique harder. For this reason, these results as well as the full features set going to be the input of the classification model that was built using different algorithms such as RandomForest Breiman (2001), REPTree Cohen (1995), LogitBoost Friedman et al. (2000), JRip Cohen (1995) and J48 Quinlan (1993).

2.5.2 Analysis and Discussion of Feature Selection Outcomes

The collected results of the different five classifiers carried out on the features sets for all datasets are generated using the data mining tool Weka and based on the testing option cross validation using 10 folds. The following discussion systematically evaluate multiple feature selection techniques across a diverse set of educational datasets. The goal

Table 2.3: Selected Attributes Using Feature Selection Techniques for "Mathematics Course Dataset"

Feature subset	Attributes	Nb of Features
Full Features Set	School, sex, age,address, famsize, Pstatus, Medu, Fedu, Mjob, Fjob, reason, guardian, traveltime, studytime, failures, schoolsup, famsup paid, activities, nursery, higher, famrel, freetime, internet, romantic, gout, Dalc, Walc, health, absences, G1, G2, G3.	33
Cfs Features	Sex, failures, G1, G2, G3.	5
Rf Features	G2, G1, Mjob, sex, Walc, Medu, paid, failures, studytime, schoolsup, famsup, Dalc, address, Pstatus, higher, famsize, internet, health, absences, Fedu, age, school, goout, G3.	24
IG Features	G2, G1, failures, Mjob, schoolsup, Fjob, higher, reason, guardian, paid, romantic, address, sex, internet, famsize, nursery, school, Pstatus, activities, famsup, G3.	21
SU Features	G2, G1, failures, higher, schoolsup, Mjob, Fjob, reason, paid, guardian, romantic, address, sex, internet, famsize, nursery, school, Pstatus, activities, famsup, G3.	21

is to determine which feature selection strategy most effectively improves model accuracy and interpretability while maintaining generalizability across domains of different sizes and complexities. These datasets represent a wide spectrum of learning contexts and sample sizes, ranging from large-scale (Language, 649 instances; Mathematics, 395 instances) to small-scale subjects (History, 19 instances; Quran, 21 instances).

In larger datasets, notably Language (649 instances) and Mathematics (395 instances), RF feature delivered robust improvements in model performance. Its ability to handle noisy and high-dimensional data allowed classifiers like Random Forest and Logit Boost to achieve higher F-measure values compared to using all features or other selection methods. Figure 2.3 and detailed results presented in Tables 2.7 and 2.8 indicate that RF feature effectively identifies and retains features that maximize discrimination among learning outcomes, thus enhancing predictive generalization. The improvement is particularly notable in educational datasets where student behavior and learning patterns often include overlapping or weakly correlated attributes.

In medium-sized datasets such as IT (95 instances), French (65 instances), and Arabic (59 instances), the RF feature algorithm continued to demonstrate strong discriminative capability as presented in Figure 2.4. For example, the Arabic dataset achieved an outstanding F-measure of 0.873 with RF-selected features using the Random Forest

Table 2.4: Selected Attributes Using Feature Selection Techniques for "Language Course Dataset"

Feature subset	Attributes	Nb of Features
Full Features Set	School, sex, age, address, famsize, Pstatus, Medu, Fedu, Mjob, Fjob, reason, guardian, traveltime, studytime, failures, schoolsup, famsup, paid, activities, nursery, higher, internet, romantic, famrel, freetime, gout, Dalc, Walc, health, absences, G1, G2, G3.	33
Cfs Features	Studytime, failures, schoolsup, paid, activities, internet, G1, G2, G3	9
Rf Features	G2, G1, Mjob, sex, Walc, Medu, paid, failures, studytime, address, schoolsup, famsup, Dalc, Pstatus, higher, famsize, internet, health, absences, Fedu, age, school, goout, G3.	24
IG Features	G2, G1, failures, higher, school, Mjob, Medu, studytime, reason, Fjob, Fedu, schoolsup, address, internet, guardian, sex, activities, paid, nursery, romantic, famsup, famsize, Pstatus, G3.	24
SU Features	G2, G1, failures, higher, school, Medu, studytime, Mjob, address, internet, guardian, sex, paid, activities, nursery, romantic, famsup famsize, Pstatus, G3.	20

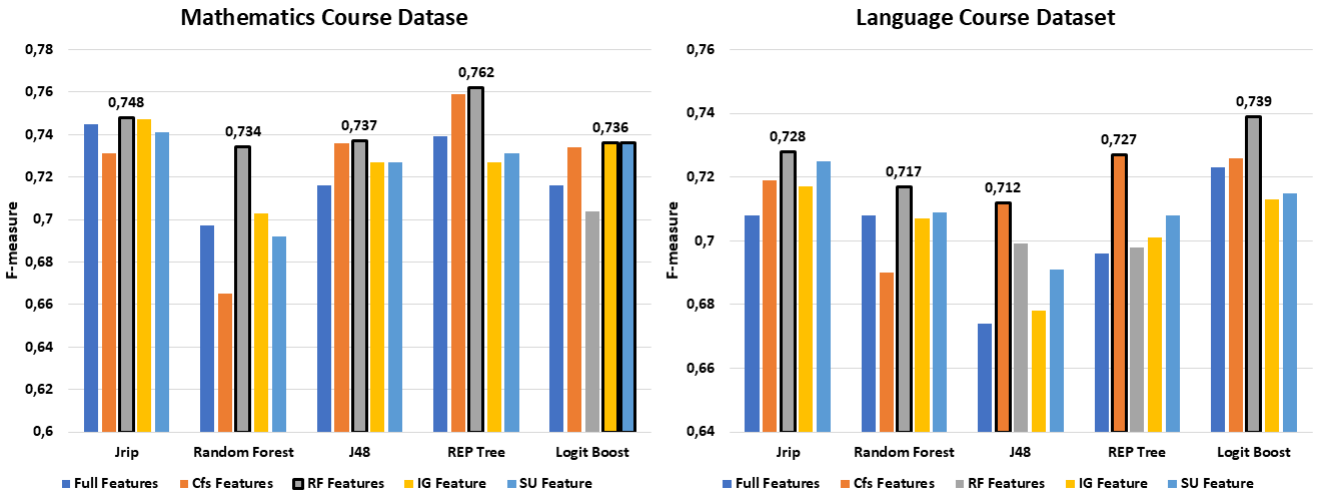


Figure 2.3: Graphical Comparison of the Performance of the Feature Selection Techniques Using the Selected Classification Algorithms

classifier, outperforming other selection methods by a clear margin. This performance stability in medium datasets reinforces the algorithm's adaptability to various data scales and its resilience to overfitting which is a common challenge when feature selection is applied to moderately sized educational datasets. ReliefF's localized evaluation of feature relevance ensures that subtle but informative attributes are retained, while redundant or noisy ones are effectively suppressed. The numerical results supporting this discus-

Table 2.5: Selected Attributes Using Feature Selection Techniques for "xAPI-Chemistry Dataset"

Full Features Set	StudentAbsenceDays, VisITedResources, PlaceofBirth, NationalITy, gender, Discussion, ParentAnsweringSurvey, raisedhands, Relation, SectionID, StageID, GradeID, ParentschoolSatisfaction, Semester, AnnouncementsView.	15
Feature subset	Attributes	Nb of Features
Cfs Features	raisedhands, VisITedResources, AnnouncementsView, StudentAbsenceDays	4
Rf Features	StudentAbsenceDays, VisITedResources, ParentAnsweringSurvey, ParentschoolSatisfaction, gender, raisedhands, Relation, AnnouncementsView, Discussion, PlaceofBirth, NationalITy	11
IG Features	VisITedResources, AnnouncementsView, StudentAbsenceDays, ParentschoolSatisfaction, ParentAnsweringSurvey, raisedhands, gender, Relation, PlaceofBirth, NationalITy	10
SU Features	VisITedResources, AnnouncementsView, StudentAbsenceDays, ParentschoolSatisfaction, ParentAnsweringSurvey, raisedhands, gender, Relation, PlaceofBirth, NationalITy	10

Table 2.6: Selected Attributes Using Feature Selection Techniques for "xAPI-English Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	GradeID, ParentAnsweringSurvey, StudentAbsenceDays	3
Rf Features	StudentAbsenceDays, GradeID, StageID, Relation, PlaceofBirth, raisedhands, ParentschoolSatisfaction, ParentAnsweringSurvey, NationalITy, AnnouncementsView, gender, VisITedResources, Discussion, SectionID	14
IG Features	StudentAbsenceDays, GradeID, NationalITy, PlaceofBirth, AnnouncementsView, StageID, raisedhands, ParentAnsweringSurvey, Relation, ParentschoolSatisfaction, gender, Semester, SectionID	13
SU Features	StudentAbsenceDays, GradeID, NationalITy, PlaceofBirth, AnnouncementsView, StageID, raisedhands, ParentAnsweringSurvey, Relation, ParentschoolSatisfaction, gender, Semester, SectionID	13

sion are presented in the Tables 2.9, 2.10 and 2.11 .

Table 2.7: Classification Results for "Mathematics Course Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0.745	0.697	0.716	0.739	0.716
Cfs Features	0.731	0.665	0.736	0.759	0.734
Rf Features	**0.748**	**0.734**	**0.737**	**0.762**	0.704
IG Features	0.747	0.703	0.727	0.727	**0.736**
SU Features	0.741	0.692	0.727	0.731	**0.736**

Table 2.8: Classification Results for "Language Course Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0.708	0.708	0.674	0.696	0.723
Cfs Features	0.719	0.690	**0.712**	**0.727**	0.726
Rf Features	**0.728**	**0.717**	0.699	0.698	**0.739**
G Features	0.717	0.707	0.678	0.701	0.713
SU Features	0.725	0.709	0.691	0.708	0.715

Table 2.9: Classification Results for "xAPI-IT Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,617	**0,571**	0,567	0,502	0,562
Cfs Features	0,614	0,614	0,571	0,527	**0,616**
Rf Features	**0.667**	0.622	0.567	0.502	0.606
IG Features	**0,617**	0,612	0,563	0,485	0,579
SU Features	**0,617**	0,612	0,563		0,485 0,579

Table 2.10: Classification Results for "xAPI-Arabic Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,692	0,822	0,777	0,704	0,758
Cfs Features	0,756	0,790	0,714	0,633	0,772
Rf Features	**0.758**	**0.873**	0.692	**0.777**	**0.786**
IG Features	0,711	0,822		**0,793** 0,688	0,704
SU Features	0,711	0,822		**0,793** 0,688	0,704

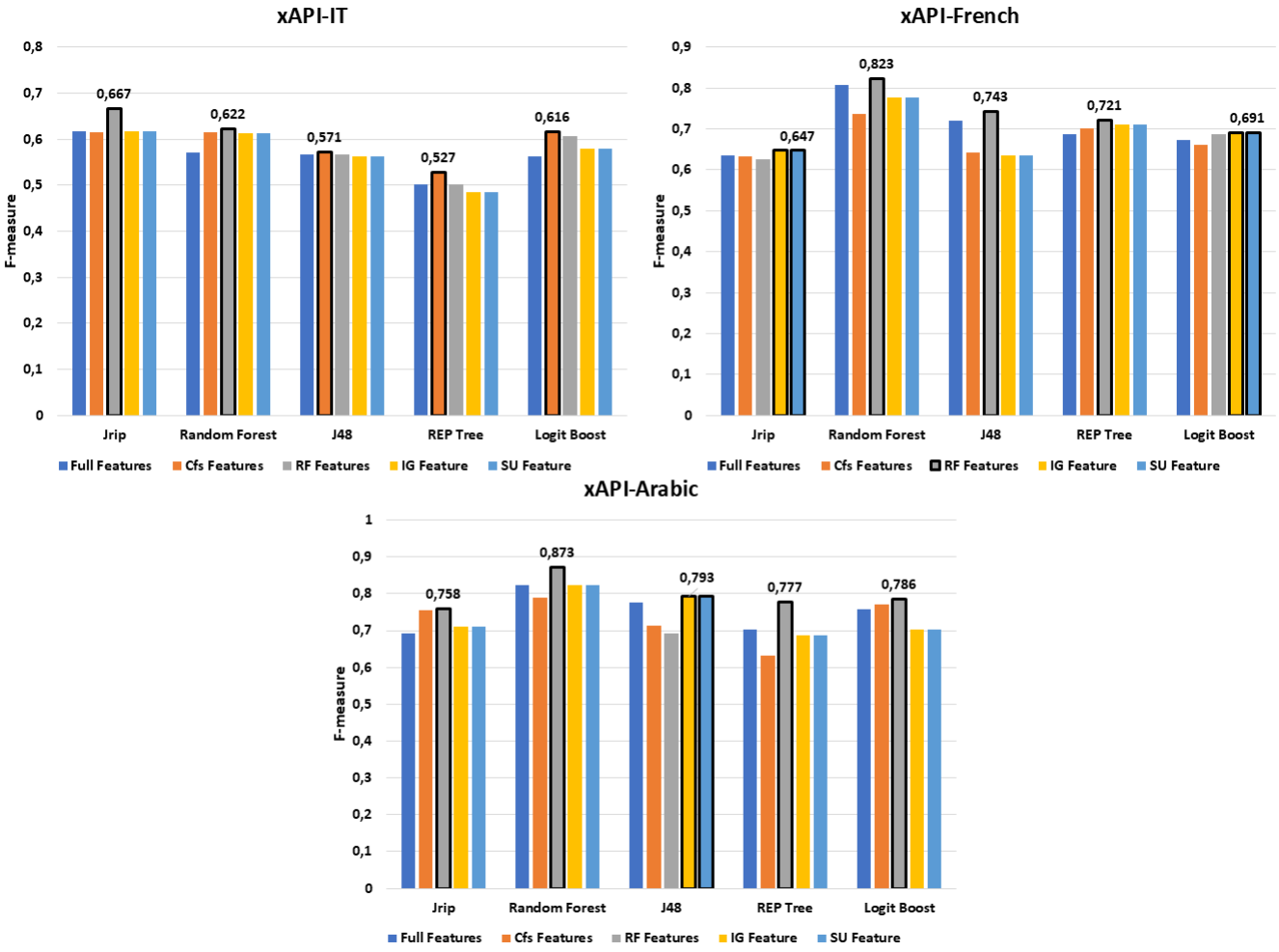


Figure 2.4: Graphical Comparison of the Performance of the Feature Selection Techniques Using the Selected Classification Algorithms

Table 2.11: Classification Results for "xAPI-French Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,634	0,807	0,721	0,686	0,672
Cfs Features	0,632	0,736	0,643	0,702	0,661
Rf Features	0.626	**0.823**	**0.743**	**0.721**	0.686
IG Features	**0,647**	0,776	0,634	0,711	**0,691**
SU Features	**0,647**	0,776	0,634	0,711	**0,691**

For small datasets such as Science (51 instances), English (45), Biology (30), Spanish (25), Chemistry (24), Geology (24), Quran (21), Math (21) and History (19), RF feature also produced highly competitive results. The detailed experimental results are summarized in tables 2.12, 2.13, 2.14, 2.15,2.16,2.17,2.18,2.19 and 2.20. As illustrated

in Figures 2.5, 2.6 and 2.7 ,in several cases, it enabled classifiers to reach F-measure values exceeding 0.8, despite the limited number of instances. This demonstrates the algorithm's robustness even when data scarcity might otherwise reduce the reliability of statistical feature selection methods. The algorithm's ability to evaluate features relative to their local instance neighborhoods appears to be particularly beneficial in such low-data conditions, ensuring that even small-scale datasets can yield meaningful and generalizable models.

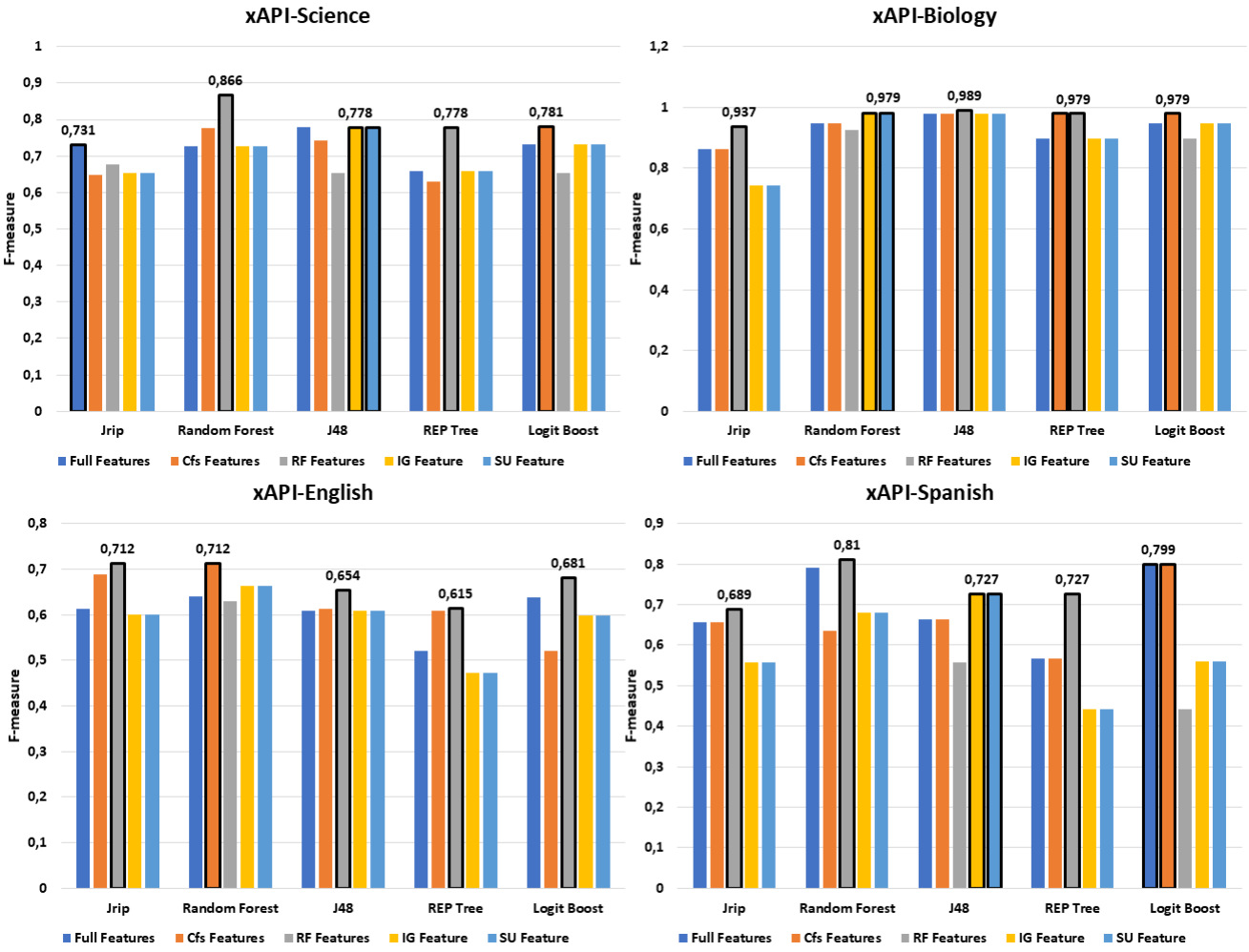


Figure 2.5: Graphical Comparison of the Performance of the Feature Selection Techniques Using the Selected Classification Algorithms

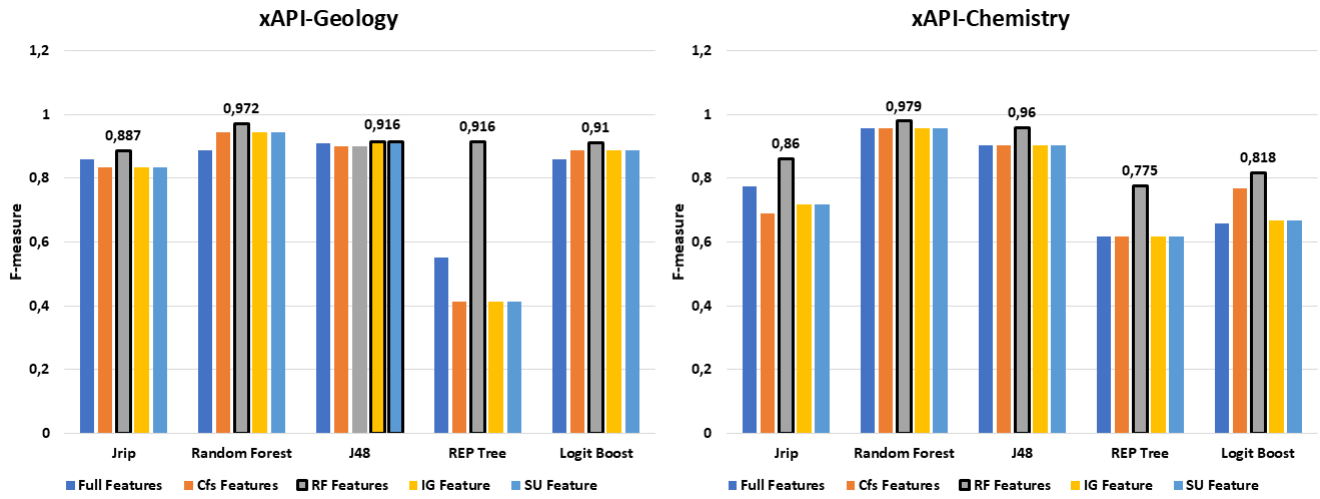


Figure 2.6: Graphical Comparison of the Performance of the Feature Selection Techniques Using the Selected Classification Algorithms

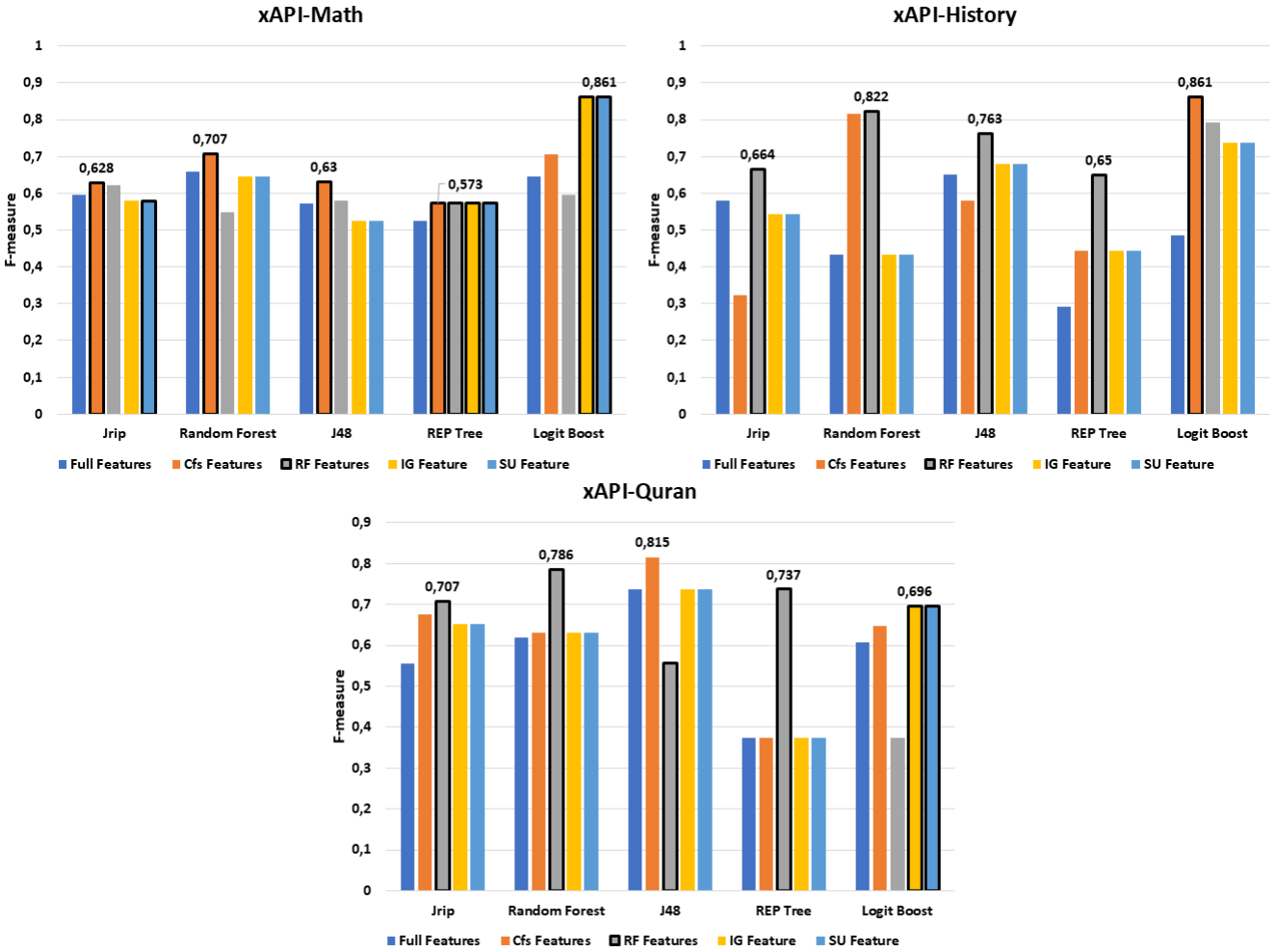


Figure 2.7: Graphical Comparison of the Performance of the Feature Selection Techniques Using the Selected Classification Algorithms

Table 2.12: Classification Results for "xAPI-Science Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	**0,731**	0,728	0,778	0,658	0,731
Cfs Features	0,648	0,776	0,743	0,629	0,781
Rf Features	0.677	**0.866**	0.653	**0.778**	**0,653**
IG Features	0,653	0,728	**0,778**	0,658	0,731
SU Features	0,653	0,728	**0,778**	0,658	0,731

Table 2.13: Classification Results for "xAPI-English Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,612	0,640	0,609	0,521	0,638
Cfs Features	0,688	**0,712**	0,612	0,609	0,521
Rf Features	**0.712**	0.629	**0.654**	**0.615**	**0.681**
IG Features	0,601	0,663	0,609	0,473	0,598
SU Features	0,601	0,663	0,609	0,473	0,598

Table 2.14: Classification Results for "xAPI-Biology Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,864	0,946	0,979	0,897	0,946
Cfs Features	0,864	0,946	0,979	**0,979**	**0,979**
Rf Features	**0.937**	0.925	**0.989**	**0.979**	0.897
IG Features	0,744	**0,979**	0,979	0,897	0,946
SU Features	0,744	**0,979**	0,979	0,897	0,946

Table 2.15: Classification Results for "xAPI-Spanish Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,657	0,790	0,663	0,568	**0,799**
Cfs Features	0,657	0,635	0,663	0,568	**0,799**
Rf Features	**0.689**	**0.810**	0.558	**0.727**	0.441
IG Features	0,558	0,681	**0,727**	0,441	0,561
SU Features	0,558	0,681	**0,727**	0,441	0,561

Table 2.16: Classification Results for "xAPI-Chemistry Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,775	0,958	0,902	0,618	0,657
Cfs Features	0,689	0,958	0,902	0,618	0,767
Rf Features	**0.860**	**0.979**	**0.960**	**0.775**	**0.818**
IG Features	0,717	0,958	0,902	0,618	0,667
SU Features	0,717	0,958	0,902	0,618	0,667

Table 2.17: Classification Results for "xAPI-Geology Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,861	0,887	0,910	0,550	0,861
Cfs Features	0,833	0,944	0,900	0,413	0,889
Rf Features	**0.887**	**0.972**	0.899	**0.916**	**0.910**
IG Features	0,833	0,944	**0,916**	0,413	0,889
SU Features	0,833	0,944	**0,916**	0,413	0,889

Table 2.18: Classification Results for "xAPI-Quran Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,556	0,620	0,737	0,373	0,607
Cfs Features	0,676	0,630	**0,815**	0,373	0,647
Rf Features	**0.707**	**0.786**	0.556	**0.737**	0.373
IG Features	0,652	0,630	**0,737**	0,373	**0,696**
SU Features	0,652	0,630	**0,737**	0,373	**0,696**

Table 2.19: Classification Results for "xAPI-Math Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,596	0,658	0,573	0,526	0,645
Cfs Features 0,707	**0,628**	0,707		0,630	**0,573**
Rf Features	0.622	0.549	**0.579**	**0.573**	0.596
IG Features	0,579	**0,645**	0,526	**0,573**	**0,861**
SU Features	0,579	**0,645**	0,526	**0,573**	**0,861**

Table 2.20: Classification Results for "xAPI-History Dataset"

Feature Subset	F-Measure				
	Jrip	Random Forest	J48	REP Tree	Logit Boost
Full Features	0,580	0,432	0,650	0,292	0,486
Cfs Features	0,324	0,815	0,580	0,444	**0,861**
Rf Features	**0.664**	0.822	**0.763**	**0.650**	0.792
IG Features	0,544	0,432	0,680	0,444	0,736
SU Features	0,544	0,432	0,680	0,444	0,736

Across all classifiers tested, RF feature consistently provided the most stable and enhanced F-measure values, particularly with ensemble learners like Random Forest and boosting algorithms such as Logit Boost. While Cfs, IG, and SU showed reasonable performance in certain domains, their sensitivity to feature correlation limited their effectiveness compared to RF feature.

In summary, results demonstrate that RF feature is the most effective feature selection technique across diverse educational datasets of varying sizes and complexity. Its instance-based approach ensures a balanced trade-off between accuracy and generalization, producing significant improvements in F-measure scores across multiple classification algorithms. Consequently, the ReliefF method can be regarded as a robust and domain-agnostic feature selection strategy for educational data mining tasks, particularly those involving heterogeneous data sources and limited sample sizes. Its ability to preserve relevant features while mitigating redundancy makes it a strong candidate for integration into predictive modeling frameworks aimed at grouping students based on learning behaviors and performance trends.

2.6 Most relevant selected attributes after applying Relief Feature algortihm

The primary objective of this thesis is to form groups of students using bio-inspired Ant Colony clustering algorithms, which require datasets to be represented in a fully numerical format Lumer and Faieta (1994), Deneubourg et al. (1992) . Since the initial xAPI-based educational datasets contain a mixture of categorical, ordinal, and continuous attributes, it is essential to preprocess and transform these heterogeneous data types into consistent numerical representations suitable for computational modeling. Accordingly, we prepare our data in order to be ready for the ACC Algorithms. First, we transform our datasets into numerical data. Therefore, all attributes should take a value

in [0..1]. For example, the attribute “ParentschoolSatisfaction” has two possible values “Good” or “Bad”, so, we categorize it into two cells (the value “Good” takes “1” and “Bad” takes 0). Then, we normalize all numerical values in order to have a common scale for all numerical attributes values in the same range [0,1]. For instance, the attribute “raisedhand” is how many times students raise their hand in class during their e-learning course. Accordingly, we use the Min-Max normalization defined by the following equation:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{2.6}$$

where x=$x_1, x_2, ..., x_n$, m is the number of instances and z_i is the i^th normalized data.

2.7 Conclusion

This chapter analyzes the impact of feature selection techniques on the classification models. We conducted a comparative study to identify the best feature selection technique for a predictive model for students’ academic performance. In this chapter, we presented the methodology we adopted for this study, where four feature selection techniques were used with five classification algorithms to determine the optimal approach to identify the best feature selection technique. Experimental results demonstrate that Rf Feature selection outperforms the other three techniques.

Chapter 3

Parameters Sensitivity Analysis of Ant Colony based Clustering

Contents

3.1 Introduction

Inspired by nature, Ant Colony based Clustering arises from ant colony behavior in organizing nests and clustering ants corpses. However, the performance of a given algorithm depends strongly on its parameters settings. Indeed, it holds a large number of adjustable parameters that need to be instantiated by suitable values. In this chapter, we study the parameters influence, more precisely the parameter α which is responsible for adjusting similarity between objects.

3.2 Parameters influence on Ants behavior based Algorithm

Parameter tuning is to find appropriate parameter settings of algorithms in order to optimize their performance. It has a strong impact on the accuracy of ant colony based clustering algorithms since it controls their behavior. Ant colony algorithms have several parameters controlling different aspects. In this section, we analyze the most important parameters as well as their influence on the algorithm performance. As illustrated in Tables 3.1, 3.2 and 3.3 we categorise these parameters depending on their related factors such as Pheromone, Ants' movement and similarity related parameters.

3.2.1 Pheromone related Parameters

Artificial ants communicate by laying synthetic pheromone along the edges on their path through a decision graph. This attracts following ants likely to search in the same region of the search space. In addition, pheromone values are used and updated by the Ant Colony algorithm during the search.
Both Chavarría-Molina et al. (2020) and Yasear and Ku-Mahamud (2021), studied the impact of the pheromone related parameters α_p, p, Q and Δ_τ on Ant colony optimization algorithm and Ant colony system. Authors in Yasear and Ku-Mahamud (2021) proposed an hybrid algorithm called Harris's hawk optimizer ant colony system (HHO-ACS) in order to improve the path finding behavior of the traditional ACS algorithm (get the optimal path). The aim of their work is to tune ACS parameters using Harris's hawk optimization in solving the traveling salesman problem. Their proposed algorithm was compared to other well known meta-heuristics such as PSO, dragonfly, GA and ACO algorithms. Similarly, Chavarría-Molina et al. (2020) proposed a hybrid clustering method named BACOK (Basic ACO improved by k-means algorithm). After extensive param-

eter fitting, their experiments show that their method performs the compared algorithm (BACO and K-means) in reasonable time on five real life datasets. In addition, they revealed that the ACO algorithm is very sensitive to p, α_p and β parameters. For instance, high values of the pheromone evaporation rate p make the pheromones evaporate faster and the past solutions are easily forgotten and the exploration of new solutions (better or worse) is encouraged. Whereas, low values make past solutions more important on the construction of new ones and helps a good solution to converge, either to a local or global optimum. On the other hand, even though they stated that the Pheromone amplification constant Q has negligible influence, it affects the convergence speed of the ACO to a certain extent. As a matter of fact, if it is large, the pheromone concentration will be highly concentrated, making the algorithm fall into a local optimum. In the same context, Sahana et al. (2019) proposed an algorithm for tuning ACO parameters (p and Δ_τ) and they proved how they affect the performance of ACO which affects in its turn the performance of grid environment when applied for scheduling. Actually, the pheromone update quantity Δ_τ aims to enhance the diversity of algorithm search and to avoid getting into local optimal. It determines the search efficiency of artificial ants, and then affects the optimization performance and evolution speed of the whole algorithm.

3.2.2 Ants movement related parameters

As we mentioned above, ACO algorithm is sensitive to some parameters such as the key parameter α_p. It is a weight assigned to pheromone concentration deposited by ants and it aims at controlling the relative importance of the pheromone trail concentrations. In other word, it reflects the strength of stochastic factors in the path search of ant colony. The greater the value, the greater the likelihood of re-choosing the path which weakened the randomness of search. Therefore, when α_p value is too large, this can also make ant colony search prematurely trapped in local optima. In fact, α_p is a complement for the parameter β and they are closely related. Their combinations are used to discuss their impact on the performance of ant colony algorithm. Indeed, β is a weight assigned to ant visibility; it reflects the strength of ant colony in priory of path search and uncertainty factors. The greater the β value, the greater the likelihood for ants to select local shortest path on a local point, although the search convergence rate speed up. However, ant colony in search of the optimal path weaken randomness to fall into local optimum easily. Actually, ant colony parameters, in literature, could have different labels. For example, Yasear and Ku-Mahamud (2021) and Sahana et al. (2019) tuned the same parameter q_0 and p_0 respectively which control the probability of ants movements between objects. Thus, they possess the same role. As suggested in the literature, good values of

this parameter tend to be close to 1 and perform best for a short run-time. In extreme cases, a value of 1 quickly leads to search stagnation. Low values of 0 generally result in better final performance and values smaller than 0.75 produce very slow convergence towards good solutions. Therefore, the strategies that decrease the speed of this parameter result in faster convergence to good solutions.

3.2.3 Similarity related parameters

Ant colony optimization and Ant colony Clustering (ACC) are two algorithms inspired from ants' behavior. The difference between both of them is described in the following: On one hand, ACO algorithm is inspired by ant's shortest food path search, ants Lay down synthetic pheromones along the edges on their path through a decision graph. This attracts following ants likely to search in the same region of the search space. Therefore, α_p parameter presents the weight assigned to pheromone concentration deposited by ants. On the other hand, ACC is arising from the nest organization of ants: building cemeteries or brood pits, clustering their corpses, sorting their larvae, etc. into several piles. Each ant starts by walking randomly around the space, thus, based on the similarity and density of the data items within the ants' local neighborhood, ants are likely to pick up items that are surrounded by dissimilar ones and tend to drop them in the cluster of similar ones. Ants distinguish their living nest-mates from dead ones through their specific corpse odor. Therefore, α parameter in the local density function presents the scaling dissimilarity parameters that allows the decision to have or not to have two items next to each other. In this context, Gao (2016) adapt ant based clustering behavior and they propose a new Abstraction Ant Colony Clustering (AACC) algorithm using data combination mechanism to improve the computational efficiency and accuracy of the AACC algorithm. Their results show that the AACC algorithm can solve the clustering problem with a high degree of accuracy and speed while providing a very good computing stability compared with ACC and K-means algorithm. They also presented a study of the main parameters that significantly affects both AACC and ACC such as the threshold for picking up object "K_p", the constant C for picking up/dropping probability parameter which can speed up the algorithm convergence if increased. During the clustering, some objects (called outliers) with high dissimilarity to all other data elements. The outliers prevent ants from dropping them, which slows down the algorithm convergence. Therefore, a larger parameter C forces the ant to drop the outliers at the later stage of the algorithm-knowing that the role of α and $\alpha 1$ is to adjust the similarity between objects and the similarity between data reactors respectively. They affect the number of clusters and the algorithm convergence rate. Authors in Gao (2016) stated

that these parameters for both AACC and ACC can be determined by trials based on sensitivity analysis results for each dataset. In addition, the influence laws of parameters for different datasets are similar. According to influence laws from the sensitivity analysis results shown in their study, the suitable values of parameters can be determined by trials: Selecting initial parameters values according to previous experiences or studies, changes them by trials and finally leads to the suitable values through some trials. As a result, the values found for the main parameters were different in each dataset. Other parameters that barely affect these algorithms can be determined through testing and experimenting and these parameters can be fixed for different datasets as for the number of objects in the data reactor visited by the current ant.
To conclude, we cannot deny the fact that almost all Ant Colony parameters influence its performance and accuracy. However, α has always been a field of studies, we can notice that almost all cited research papers have studied the value of α due to its importance and influence on the algorithm performance. For this reason, in this paper we present a study of the key parameter α and its sensitivity when applying ACC algorithm for constructing collaborative learning teamwork.

3.2.4 Alpha (α) parameter sensitivity analysis

As stated earlier, bio-inspired algorithms are sensitive to their parameters' settings, as they influence on the clustering robustness and performance. In this research work, we pinpoint the key parameter α since it plays an important role in Ant Colony Clustering algorithms. In fact, α presents the scaling dissimilarity parameters that determine when two items should, or should not, be located next to each other. As to say, α adjusts the similarity between objects and determines the percentage of items on the grid that are classified as similar. Therefore, a too large choice of α leads to the fusion of individual clusters, and in some cases, all items could be gathered within one cluster. Nontheless, a too small choice of α could prevent the formation of clusters on the grid Boryczka (2008).
Accordingly, α affects the number of clusters and the algorithm convergence rate. Objects with greater degrees of similarity have greater values of α and tend to cluster. Thus, the number of clusters decreases, and the algorithm becomes faster. On the contrary, if α is smaller, the objects have smaller degrees of similarity, and the larger group will split into smaller groups. Thus, the number of clusters will increase, and the algorithm will become slowerGao (2016). Therefore, during the experimental studies, we notice that α affects clearly the clustering results. This paper studies the effect of α parameter on different dimensional Educational datasets.

Table 3.1: Review summary on ant based clustering algorithms parameters: Pheromone related parameters

Parameters	**Ants Behavior Inspired Algorithms**		**Description**	**Parameters Range**	**Ref**
	ACO	**ACS**			
α_p	X	X	Weight assigned to pheromone concentration deposited by ants	[0, 0.25, .., 6]	Chavarría-Molina et al. (2020)
				{0.8848, 0.8929, 0.9419, 0.9822, 1.0327, 1.0678, 1.1288, 1.1919}	Yasear and Ku-Mahamud (2021)
p	X	X	Pheromone evaporation rate	[0.1, .. , 0.9]	
				{0.0993, 0.0994, 0.0995, 0.997, 0.0999}	
Q	X		Pheromone amplification constant	[50, 100, 150, .., 500]	Chavarría-Molina et al. (2020)
Δ_τ	X		Deposited pheromone of ants	[0.00155..0.00161]	Sahana et al. (2019)

Table 3.2: Review summary on ant based clustering algorithms parameters: Ant's movement related parameters

Parameters	**Ants Behavior Inspired Algorithms**		**Description**	**Parameters Range**	**Ref**
	ACO	**ACS**			
β	X	X	Weight assigned to ant visibility	[0, 0.25, .., 6]	Chavarría-Molina et al. (2020)
				{ 1.9904, 1.9912, 1.9914, 1.9926, 1.9942, 1.9957, 1.9983, 1.9995}	Yasear and Ku-Mahamud (2021)
q_0	X	X	Control the probability of ant's movements between objects	{0.8907, 0.8909, 0.8938, 0.8941, 0.8942, 0.8944, 0.8957, 0.8970}	
p_0				[0.1467..0.1473]	Sahana et al. (2019)

Table 3.3: Review summary on ant based clustering algorithms parameters: Similarity related parameters

Parameters	**Ants Behavior Inspired Algorithms**		**Description**	**Parameters Range**	**Ref**
	ACC	**AACC**			
α	X	X	Adjust the similarity between objects	{0.45, 0.5, 0.6, 0.7, 1.5}	Gao (2016)
$\alpha 1$		X	Adjust the similarity between data reactors	{0.25, 0.3, 0.4, 0.5}	
k_c		X	Threshold for similar data reactor combination	{0.05, 0.15, 0.25}	
C	X		Constant for picking up/dropping probability	{3, 5, 6, 8}	
k_p		X	Threshold for picking up one data object.	{0.05, 0.1, 0.2}	

3.3 Ant colony based clustering for student group formation

Our research work process is illustrated in Figure 3.1. First, we start with data preparation and selection of relevant attributes Abid et al. (2017). Then, we select one of the above mentioned ACC algorithms (L&F, ACA, ACAM or Improved ACA). After setting their parameters, we run each algorithm for every 0.01 step between 0,5 and 0,6. All F-measure results are recorded and saved in order to be studied and discussed.

PseudoCode: α parameter selection process

Input : K Educational Datasets;
inf← 0.5;
sup← 0.6

Output : α with the best F-measure

```
Foreach (dataset ∈ K ) do
    Data preparation and selection of relevant attributes;
    Foreach ACC Algorithms ∈ {L&F; ACA; ACAM; Improved ACA} do
        Initializations of parameters;
        while (inf <=α<=sup) do
            Ants are randomly initialized to data;
            Compute the local density function f (data_element) by eq.3;
            Ants move data to the suitable cluster or create a new one;
            Output of the clustering results and recording F-measure;
            α← α + 0.01
        end
        Comparing results and selecting α with the best F-measure;
    end
end
```

Figure 3.1: Pseudo Code of our α parameter selection process

3.3.1 Educational Datasets Description

Our experimental benchmark is represented by 14 concrete educational datasets from the literature. These datasets are detailed as follows: Two datasets constructed by Cortez et al. Cortez and Silva (2008b), each one has 33 attributes presenting students' academic and social data for a specific core classes: Mathematics and Portuguese language; a package of other dataset called xAPI-Edu-Data collected by Amrieh et al. Amrieh et al. (2015) and Amrieh et al. (2016). This latter dataset consists of 480 students' records over 12 different subjects. Since we are grouping learners into teams to study together

the same subject, it is required to arrange this dataset into subsets according to the topic of the course. Accordingly, we obtain 12 different datasets which are: Arabic, Spanish, IT (Information Technology), Biology, Chemistry, English, French, Geology, History, Math, Quran and Science.

The first step of our process is to prepare and improve the quality of the data. This part of our research work is already presented and detailed in a previous publication Abid et al. (2017) and detailed in Chapter 2.

3.3.2 Ant Clustering algorithms parameters settings

To be well executed, some parameters and conditions of the selected algorithms has to be initialized. Hence, Table 3.4 presents the initialization parameters of the four Ant Clustering algorithms.

As mentioned previously, the value of α is data dependent and it is determined experimentally by repetitive executions of each ACC algorithm. In fact, the key parameter α has attracted several researchers and has always been under study because of its strong influence on Ant Colony Algorithms' performance. After several simulation experiments, we notice that the best accuracy value in most cases is when α is between 0.5 and 0.6. So, we investigate the influence of α for every 0.01 in the range of [0.5 .. 0.6]. Thus, Table 3.5 presents the α values elected for each algorithm per dataset. As for xAPI-French dataset, the selected α values in regards to L&F, ACA, ACAM and Improved ACA are 0.54, 0.55, 0.57 and 0.54 respectively; since they show the best F-measure value for each clustering algorithm.

Table 3.4: The algorithms parameters used during the simulation. These parameters are randomly generated according to Boryczka (2009) and Gao (2016) studies

Parameters	**Algorithms**			
	L&F	**ACA**	**ACAM**	**Improved ACA**
γ_{pick}	0.1	-	-	-
γ_{drop}	0.15	-	-	-
σ	5	5	-	5
Memory	-	8	8	-
Max speed	-	-	-	20
Sigmoid(f(o_i))	-	-	-	c=5
Number of ants	10			
Number of Iterations	20,000			
Neighborhood radius	9			
α	See TABLE 3.5			

Table 3.5: α values derived after multiple explorations per dataset

DataSets	**α Value**			
	L&F	**ACA**	**ACAM**	**Improved ACA**
xAPI-IT	0.5	0.5	0.57	0.57
xAPI-Spanish	0.55	0.52	0.5	0.54
xAPI-Arabic	0.55	0.51	0.54	0.51
xAPI-Biology	0.5	0.51	0.55	0.59
xAPI-Chemistry	0.5	0.58	0.51	0.6
xAPI-English	0.53	0.5	0.52	0.53
xAPI-French	0.54	0.55	0.57	0.54
xAPI-Geology	0.55	0.56	0.5	0.56
xAPI-History	0.59	0.51	0.5	0.55
xAPI-Math	0.54	0.51	0.5	0.51
xAPI-Quran	0.5	0.56	0.5	0.59
xAPI-Science	0.57	0.55	0.52	0.56
Mathematics Course	0.52	0.5	0.57	0.57
Portuguese language Course	0.57	0.56	0.59	0.59

3.3.3 Ant Clustering algorithms for learners' grouping

As an instance of our clustering evolution, L&F algorithm simulation is illustrated in Figure 3.2, where datasets items are randomly scattered on the grid. By running the algorithm, 10 artificial ants start exploring the environment with random movement along the grid. Then, each one select data point and move it toward its new group based on

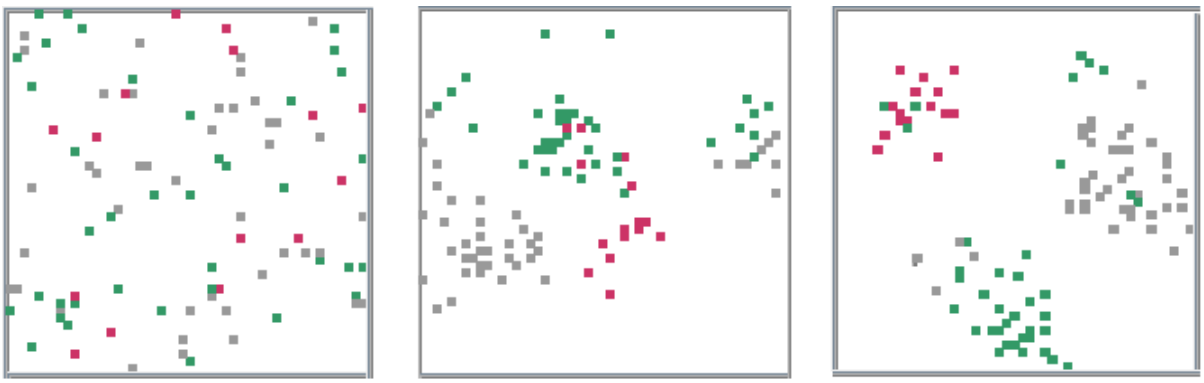


Figure 3.2: Simulation of L&F clustering algorithm at (a) start, (b) iteration 10,000 and (c) iteration 20,000

grade similarities during 20,000 iterations. For performance reasons, the number of ants

should be kept small. Since ants walk randomly on the grid, too many ants should not have any effect, i.e., two ants coincide many times, over and over again, but they follow different walk Boryczka (2009). Figure 3.2 illustrates three states of L&F algorithm: At the start point, at the 10,000th iteration and the last point of the 20,000th iteration.

3.3.4 Performance metric

We evaluate the clustering accuracy of the four selected ACC algorithms based on F-measure, which is bound by the interval [0..1]. It represents the harmonic mean between the precision and recall of the clustering for all classes presented as following:

$$F_{combined} = \sum_{i} \frac{n_i}{n} max_j F(i,j) \tag{3.1}$$

Where:

- n: the total number of objects being clustered.
- n_j : the number of objects in class i.
- $maxF(i,j)$: the maximum F-measure that each class sees over all clusters given by:

$$F(i,j) = \frac{2Precision(i,j)Recall(i,j)}{Precision(i,j) + Recall(i,j)} \tag{3.2}$$

Where:

- $Precision(i,j)$: the proportion of items in cluster j that are of class i given by Equation 3.3.
- $Recall(i,j)$: the proportion of class i that belongs to that cluster j given by Equation 3.4.

$$Precision(i,j) = \frac{n_{ij}}{n_j} \tag{3.3}$$

$$Recall(i,j) = \frac{n_{ij}}{n_i} \tag{3.4}$$

Where:

- n_{ij}: the number of objects of class i within cluster j.

3.4 Results Discussion

This section presents in details the conclusions draw from the experimental results after applying the ant algorithms to a benchmark consisting of 14 Educational datasets provided by Cortez and Silva (2008b), Amrieh et al. (2015) and Amrieh et al. (2016).

3.4.1 Evaluation of the Ant based clustering Algorithms Performance based on F-measure

In this section we investigate the performance of the four selected ant colony based clustering algorithms in educational fields. As depicted by Figure 3.3, the reported results can give us a rough estimation about the effectiveness of the ant algorithms. The obtained radar diagram visualises the results presented in Table 3.6, obtained at the end-run of 20.000 iterations. This big number of iterations is a common characteristic of different ant-based clustering algorithms, and even more for any heuristic method.
Therefore, we can conclude that ACAM algorithm outperform the others in most of the tested datasets. Although, we can notice that it gives good results especially using datasets with reduced number of instances (between 19 and 95 instances), while Improved ACA algorithm proves that it can handle datasets with much more instances (395 and 649 instances).

3.4.2 Sensitivity analysis of α parameter for ACAM and Improved ACA algorithms

Based on the above results, we choose to deepen our analysis about the sensitivity of α parameter for ACAM and Improved ACA algorithms since they outperform L&F and ACA algorithms. In fact, we analyze the impact of α value on the F-measure of ACAM with xAPI datasets and Improved ACA with Mathematics/Portuguese course datasets.
Figures 3.4, 3.5, 3.6, 3.7 and 3.8, illustrate a potential correlation, nearly to a linear relationship, between α values and the performances of the ACAM and Improved ACA algorithms. On one hand, Figures 3.4, 3.5, 3.6 and 3.7 present the variation of F-measures according to the growing values of α from 0.5 to 0.6 for xAPI dataset. We can notice that there is a negligible influence of α on the ACAM algorithm performance. On the other hand, as illustrated by 3.8, linear function of Portuguese course dataset, which is almost double sized than the Mathematics one, shows that when α increase the F-measure

Table 3.6: Comparative F-measure results of L&F, ACA, ACAM and Improved ACA algorithms

Datasets	L&F	ACA	ACAM	Improved ACA
xAPI-IT	0.79	0.9	**0.92**	0.8
xAPI-Spanish	0.61	0.55	**0.97**	0.58
xAPI-Arabic	0.71	0.94	**0.98**	0.49
xAPI-Biology	0.52	0.6	**0.98**	0.53
xAPI-Chemistry	0.57	0.62	**0.97**	0.5
xAPI-English	0.72	0.75	**0.98**	0.53
xAPI-French	0.68	0.89	**0.96**	0.49
xAPI-Geology	0.8	0.74	**0.95**	0.74
xAPI-History	0.57	0.56	**0.99**	0.58
xAPI-Math	0.57	0.65	**0.93**	0.51
xAPI-Quran	0.56	0.54	**0.97**	0.59
xAPI-Science	0.72	0.77	**0.92**	0.57
Mathematics Course	0.71	0.47	0.59	**0.79**
Portuguese language Course	0.49	0.37	0.59	**0.94**

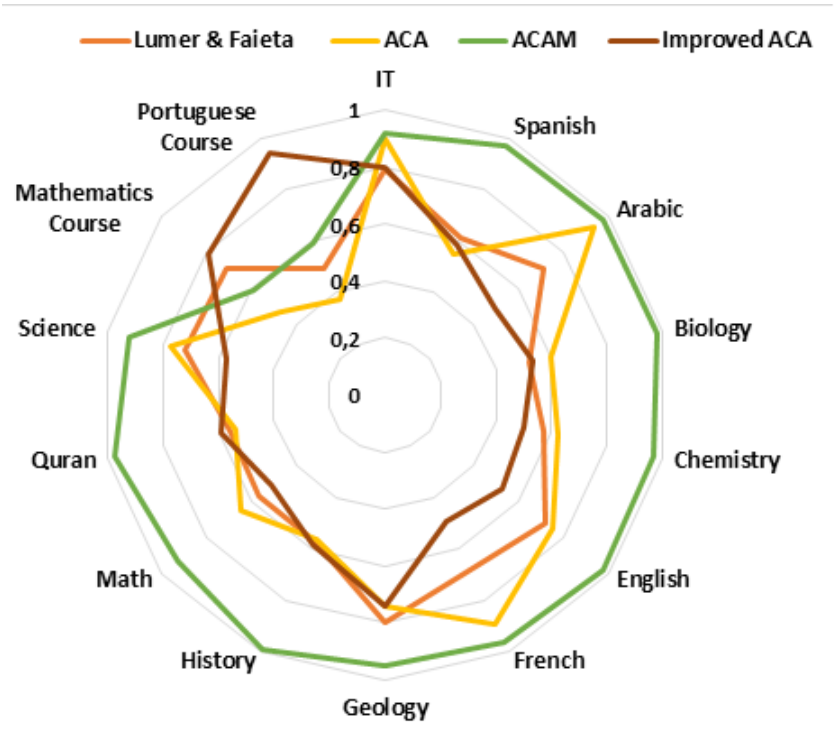


Figure 3.3: Radar analysis performances of L&F, ACA, ACAM and Improved ACA algorithms applied to 14 educational DataSets

decrease. While, for the Mathematics dataset, we notice that as long as α increase, F-measure increase too. Thus, these findings prove that as long as the datasets dimension increase, the influence of α parameter on the performance of ACC algorithms, increase too.

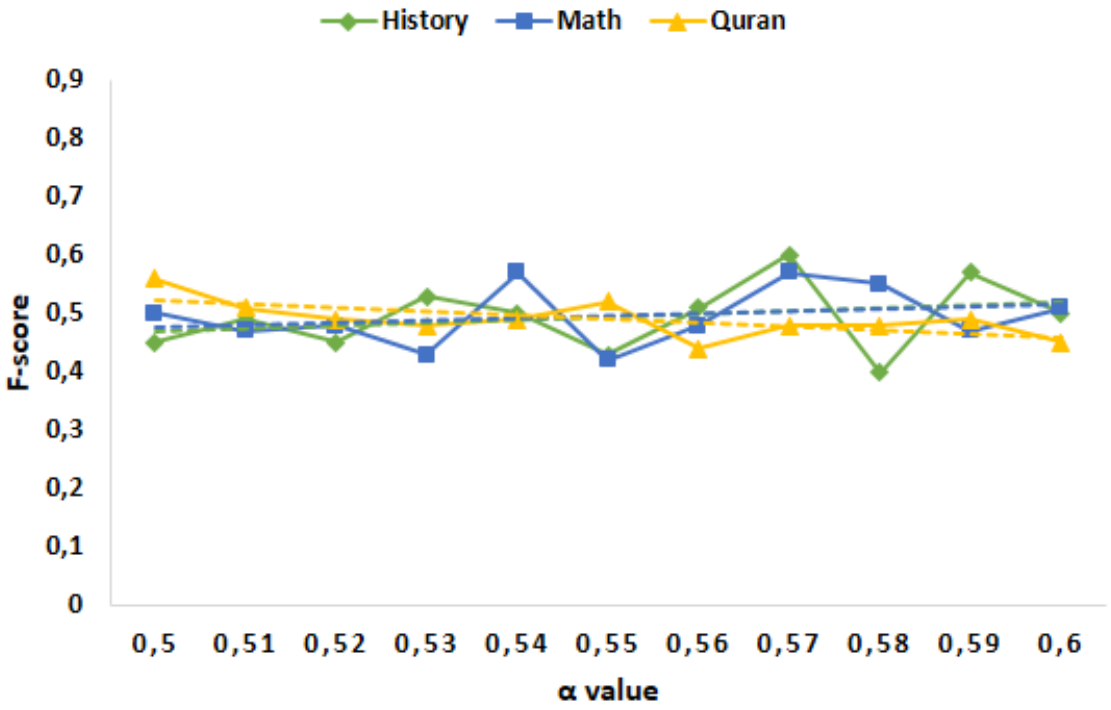


Figure 3.4: α sensitivity for Improved ACA algorithm applied on History, Math and Quran xAPI-Datasets

3.4.3 Relationship between α value and attribute selection

As stated in Abid et al. (2017), selecting relevant attributes and reducing redundant/irrelevant features improve the performance of the algorithms. Accordingly, in this section, we investigate the relationship between α value and attribute selection technique. We carried out a series of Improved ACA algorithm running on both Mathematics and Portuguese course datasets before and after attribute selection for each value of α between 0,5 and 0,6. As illustrated in Figure 3.9, the performance of the algorithm does not show any difference whether we applied RF algorithm for selecting relevant attributes or not. In fact, ACC algorithms accuracy is barely influenced by α value when applying on small datasets (i.e. xAPI-Datasets). As results, feature selection step could be ignored since it has a negligible influence on the algorithm performance even with a different value of α.

However, datasets with large number of features lead to high computational complexity

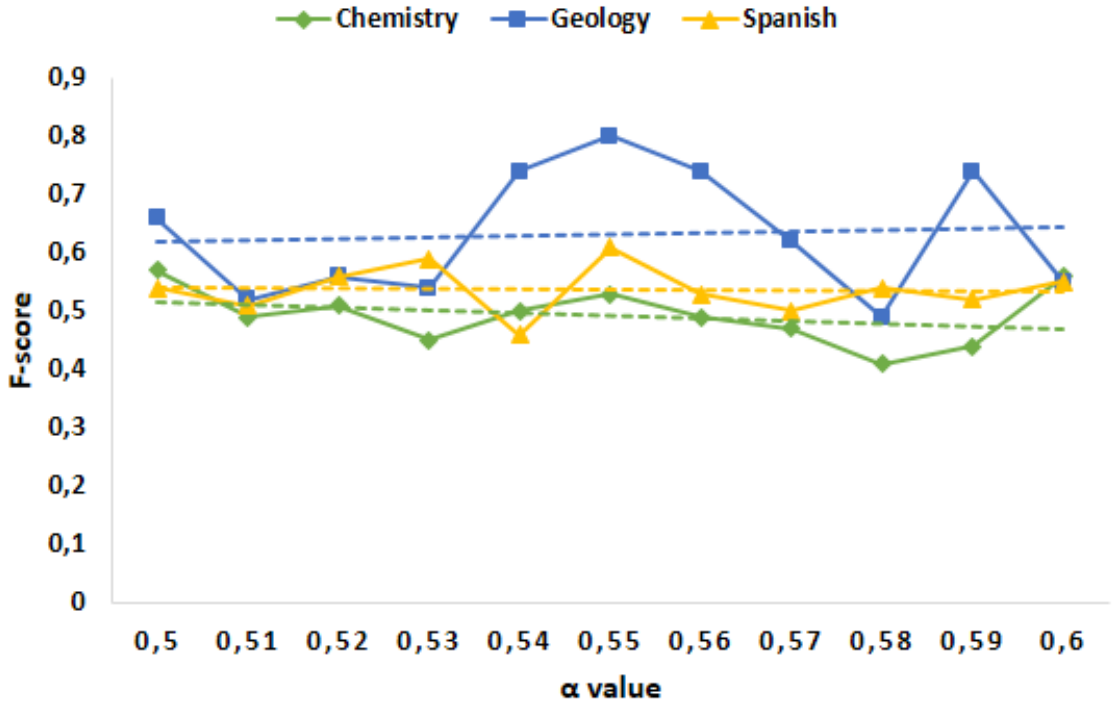


Figure 3.5: α sensitivity for Improved ACA algorithm applied on Chemistry, Geology and Spanish xAPI-Datasets

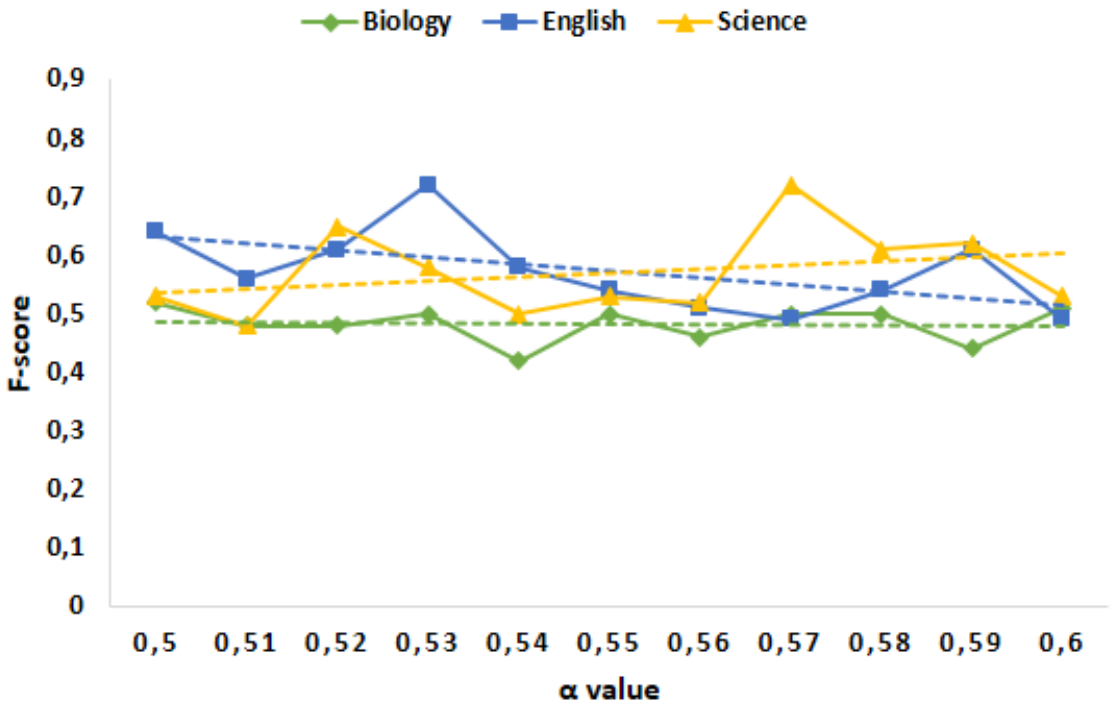


Figure 3.6: α sensitivity for Improved ACA algorithm applied on Biology, English and Science xAPI-Datasets

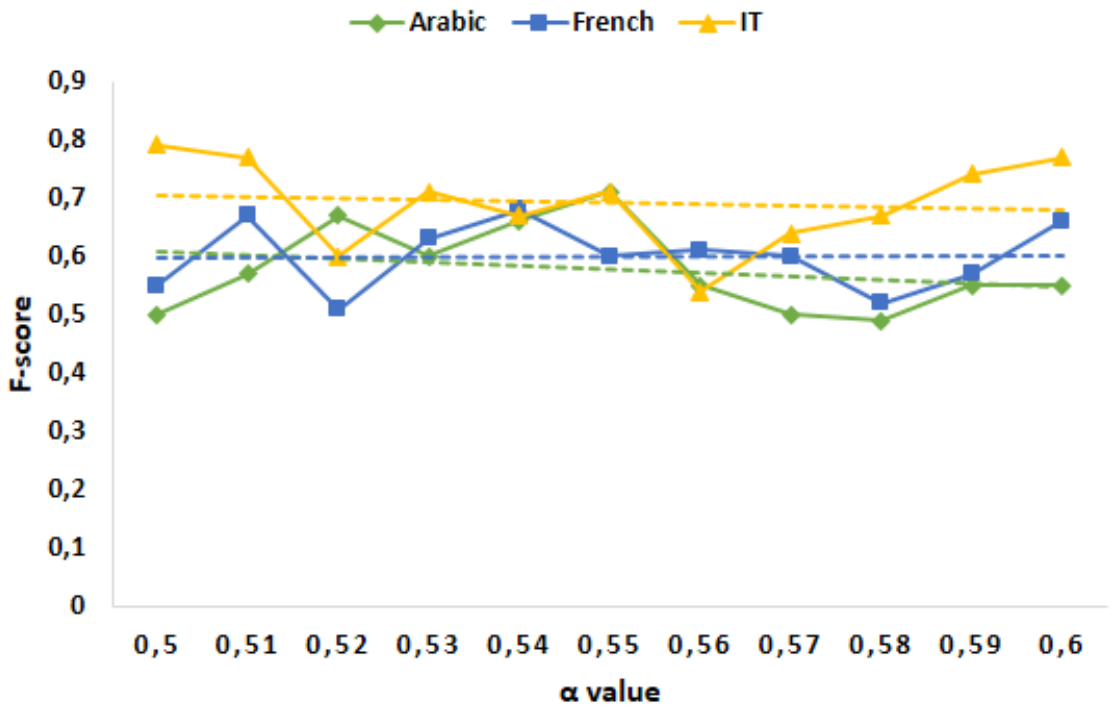


Figure 3.7: α sensitivity for Improved ACA algorithm applied on Arabic, French and IT xAPI-Datasets

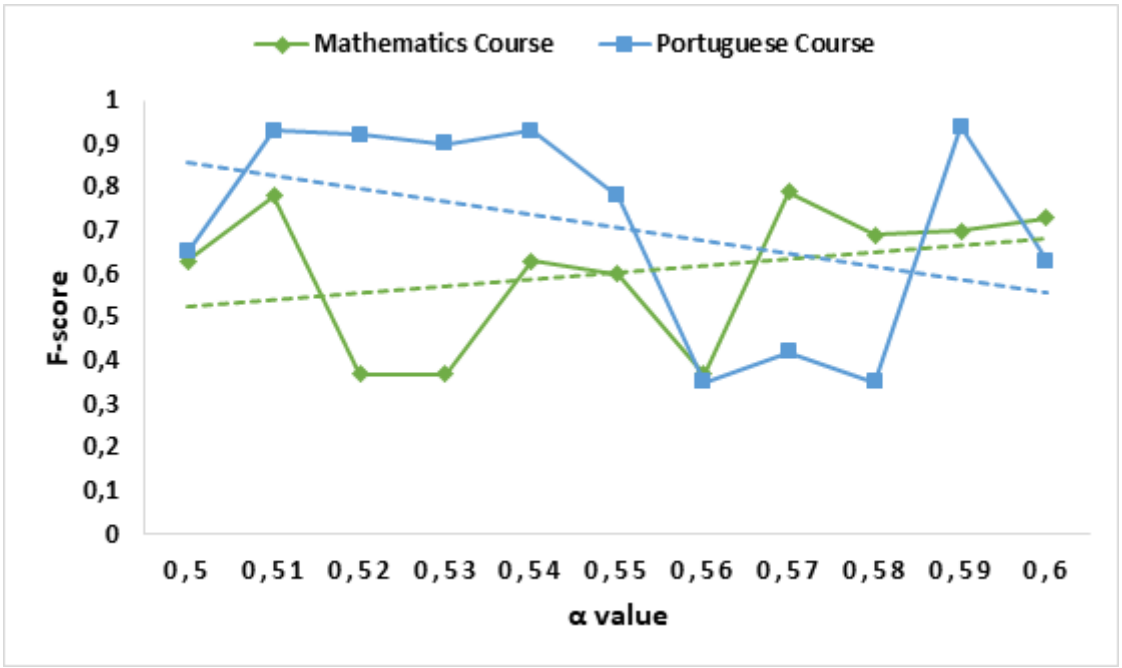


Figure 3.8: α sensitivity for Improved ACA algorithm applied on Mathematics and Portuguese Datasets

for ACC algorithms Chniter et al. (2022). Thus, further analyses are needed to investigate this issue especially to find the best trade-off between run-time complexity and clustering performance. Whereas, the findings presented in Figure 3.10 show that se-

lecting appropriate features do affect the performance of the Improved ACA algorithm when applying to the Portuguese course datasets. In addition, when applying the algorithm on the subset, the best accuracy is when α equals 0.59. However, using the full-set dataset the value of α for the best F-measure is 0.53.
Therefore, the fact that stands out from these findings is that α parameter has a remarkable and a strong influence on Ant based Clustering algorithm performance especially for high dimensional datasets.

Figure 3.9: Improved ACA performance with and without feature selection applied on Mathematics Datasets for different α values

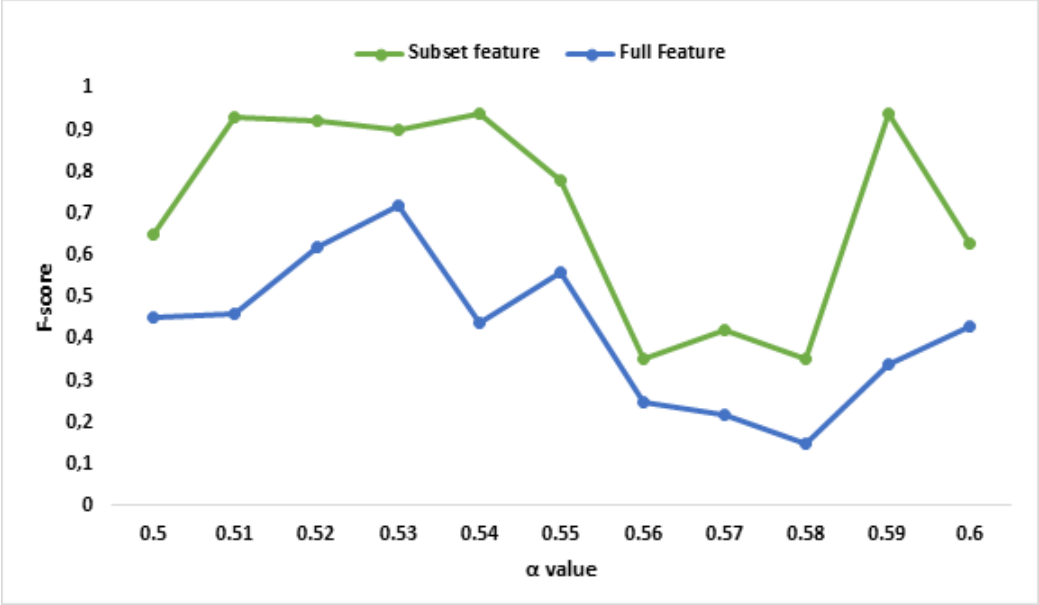


Figure 3.10: Improved ACA performance with and without feature selection applied on Portuguese Datasets for different α values

3.5 Conclusion

This chapter focused on examining the sensitivity of the α parameter in ant colony–based clustering algorithms, particularly within the context of forming student groups for collaborative learning, as discussed by Abid et al. (2023). Through extensive experimentation on both large and small educational datasets, the study highlighted that the α parameter plays a crucial role in shaping algorithmic performance. The results show that while the performance of Ant Colony algorithms is deeply influenced by α, it also varies depending on the dataset's scale. Specifically, the Improved ACA demonstrates superior adaptability and efficiency when handling large datasets, whereas the ACAM algorithm yields more accurate and stable clustering outcomes on smaller datasets. Although certain algorithms exhibited limited efficiency outside their optimal dataset sizes, these findings open promising perspectives for further optimization rather than suggesting strict constraints. The observed sensitivity underscores the flexibility of Ant Colony algorithms and their potential for fine-tuning to suit diverse educational contexts. Future research could explore dynamic adaptation mechanisms for α or hybrid models that balance performance across dataset scales. Overall, this investigation contributes valuable insights into parameter optimization, enhancing the applicability of ant colony–based clustering in collaborative learning environments.

Chapter 4

Ant Based Clustering Approach For Building Collaborative Learning Teams

Contents

4.1 Introduction

This chapter focuses on addressing one of the most challenging problems in clustering: the assignment of items that cannot be grouped into any cluster. We propose a new method of hybrid Bio-inspired Ant Colony Clustering for Collaborative Learning Teams to create synergistic groups of students. This method integrates the k-means algorithm with ant colony clustering algorithms. Initially, ant-clustering algorithms automatically create clusters without prior knowledge of the number of clusters. Subsequently, the k-means algorithm uses the total within-cluster sum of squares (WCSS) to measure homogeneity within clusters and determine the optimal number of clusters. The hybridization of these two algorithms resolves the issue of ungrouped students by assigning them to existing clusters or creating new ones as illustrated in Figure 4.1

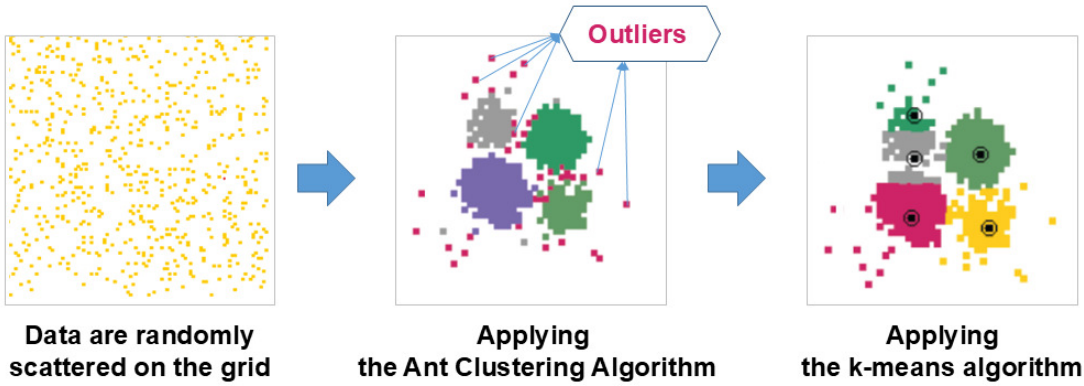


Figure 4.1: Visualization of the impact of hybridization of supervised and unsupervised methods

4.2 Hybrid K-means and Ant-based Clustering Algorithm

Ant colony clustering algorithms are categorized as unsupervised learning model. Inspired by the collective behavior of ants, which has been refined through millions of years of evolution, making them inherently robust and effective in solving complex optimization problems. These algorithms offer several advantages such as their efficiency in finding good clustering solutions, especially in large, non-linearly, high-dimensional and complex datasets, etc. Furthermore, these algorithms are not restricted to specific data types or clustering criteria and can be applied to various types of data and clustering objectives. Their adaptability to incorporate domain-specific knowledge as we can mention their application in a large variety of application fields, such as Road Traffic Management Kammoun et al. (2011), electricity-theft detection Yang et al. (2024)

and students' academic performance analysis Xu and Kim (2024). In addition, these algorithms are robust to dynamic environments where they can adapt to changes in the dataset or problem domain by continuously updating the clustering solution. Although all the mentioned above advantages, the stochastic nature of ant colony clustering algorithms allows them to escape local optima and explore a diverse range of clustering solutions and they are counted as non deterministic algorithms which mean that some items my not be assigned to a cluster. In our case, since we apply ant clustering algorithms on educational settings, students should not be considered as outliers and they must be assigned to a group of learners. As a result, we deeping our research study in order to find a solution which guarantee that all learners belongs to a specific group. Among several deterministic algorithms, we refer to the idea of integrating the K-means algorithm in order to achieve convergence. This choice is justified by its computational efficiency, time complexity and scales with large datasets. In addition, K-means algorithm is known by its robustness to outliers where its reliance on the mean (centroid) makes it relatively robust to small perturbations and noise in the data. However, despite all these advantages, it's worth to mention that K-means has sensitivity to the initial choice of centroids and the requirement of specifying the number of clusters, and the assumption of isotropic clusters with equal variance. Furthermore, it may not perform well on non-linear or irregularly shaped clusters. To bring it together, K-means algorithm is a quite good fit to Ant colony clustering algorithms and that thanks to the filling of the disadvantages of one by the advantages of the other. Therefore, Figure 4.2 presents the Hybridization of Ant-based Clustering Algorithm with the determinist K-means algorithm flowchart to guarantee the complete convergence of the students grouping.

4.3 Hybrid ant based clustering algorithm: Experimental Environment

To simulate our hybrid clustering method, we develop a simulator environment using JAVA programming language. Our purpose is to extract and test the resulting groups showing an interesting starting point for further works. We take our inspiration from Boryczka (2009) study and its cited papers like Handl et al. (2003) to follow the evaluation measure that ensure the clustering process and its validity.

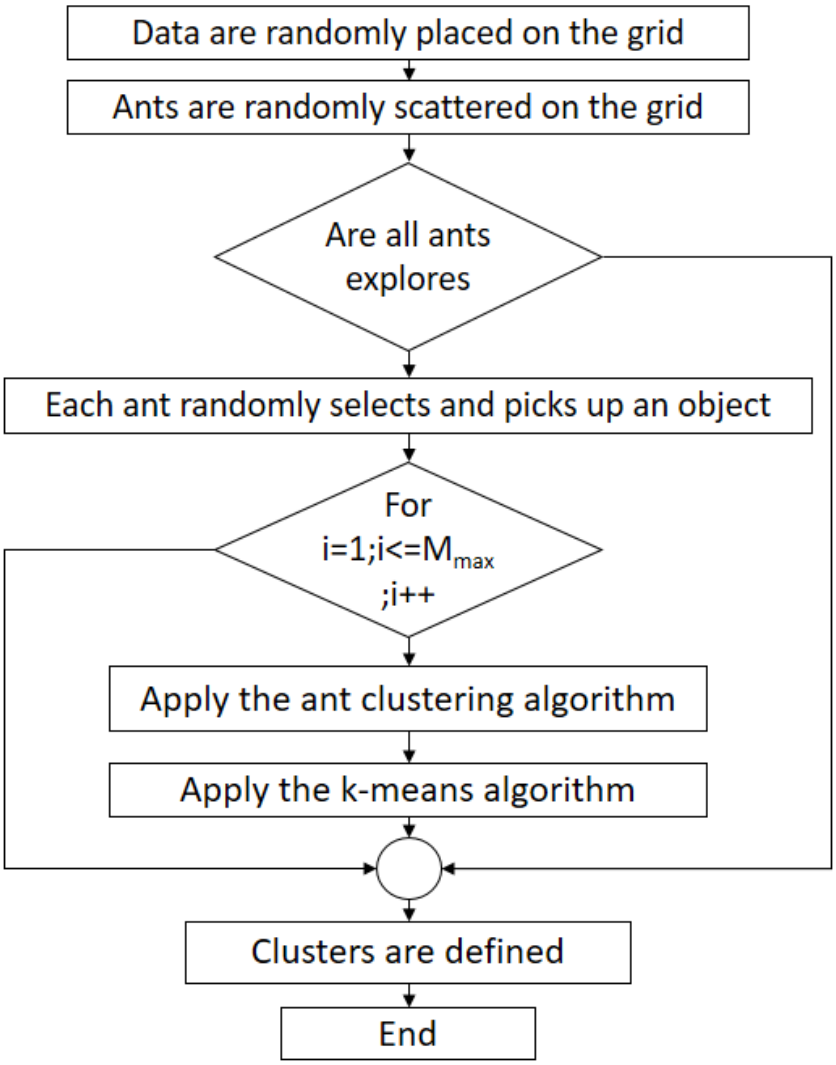


Figure 4.2: KM-AC Algorithm Flowchart

4.3.1 Data Sets Description

We tested our system on 14 concrete educational data sets. These datasets are defined as follows: Two datasets provided by Cortez and Silva (2008a), each one has 33 attributes presenting students' academic and social data for a specific core classes: Mathematics and Portuguese language. In addition, we also tested our system using xAPI-Edu-Data collected by Amrieh et al. (2015) Amrieh et al. (2016). This datasets consists of 480 student records over 12 different subjects. Since we are grouping learners into teams to study together for the same lecture, we divided this dataset into subsets according to the topic of course. It has 12 different subjects which are: Arabic, Spanish, IT, Biology, Chemistry, English, French, Geology, History, Math, Quran and Science. Therefore we got 12 subsets from the xAPIEdu-Data. The first step of our process is to prepare and improve the quality of the data. This part of our research work is already presented and

detailed in a previous publication Abid et al. (2017) and detailed in **Appendix 1**.

4.3.2 Simulation Parameters

The figure 4.3 presents the set of simulation parameters:

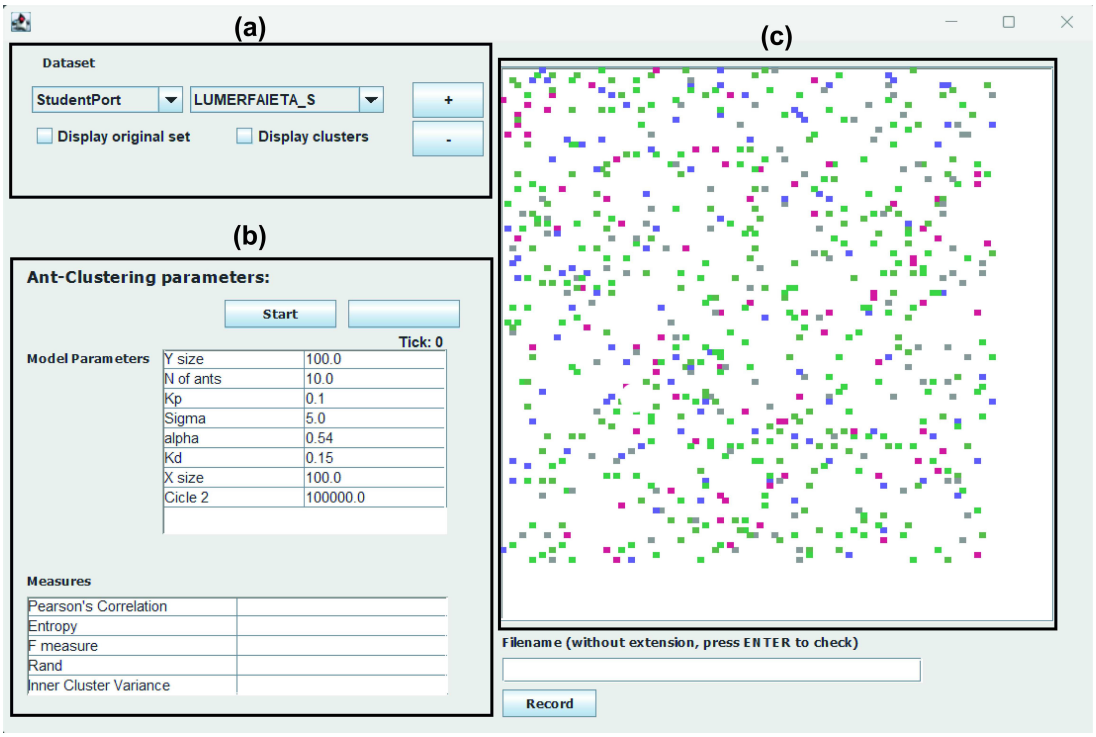


Figure 4.3: Interface of initialization parameters of Our Hybrid Method:
(a) The initialization of students parameters
(b) The Ant-Clustering parameters and their measures
(c) The Simulation graph

(a) shows the initialization parameter of data set (Language /Mathematic subject) and the simulated algorithm.

(b) shows the ant-based clustering parameters:

- 10 artificial ants are assigned to sort and move individuals.
- 0.1 is assigned for the constant threshold of pick up and 0.15 for the constant threshold of drop.
- the scaling parameter α is equal to 0.82.
- neighbor size is equal to 5.
- 100000 iterations are initialized during the run process.

Table 4.1: The algorithms parameters used during the simulation. These parameters are randomly generated according to Boryczka (2009) and Gao (2016) studies

Algorithm	**Parameters**						
	α	γ_{pick}	γ_{drop}	**memory**	**max speed**	σ	**Sigmoid (f(o_i))**
L&F	see Table 4.3	0.1	0.15	-	-	5	-
ACA		-	-	8	-	5	-
ACAM		-	-	8	-	-	-
Improved ACA		-	-	-	20	5	c=5

These parameters are changing according to the selected algorithm as shown in table 4.1

- ant memory size equal to 8 for ACA and ACAM.
- max speed of each ant equal to 20 for the Improved ACA.

Then, the simulation phase is showed in (c) to be able to see the individuals relocation along the grid and the groups building. In the initialization step, the individuals are randomly scattered on the grid.

Table 4.2: α values derived after multiple explorations per dataset Abid et al. (2023)

DataSets	α **Value**			
	L&F	**ACA**	**ACAM**	**Improved ACA**
IT	0.5	0.5	0.57	0.57
Spanish	0.55	0.52	0.5	0.54
Arabic	0.55	0.51	0.54	0.51
Biology	0.5	0.51	0.55	0.59
Chemistry	0.5	0.58	0.51	0.6
English	0.53	0.5	0.52	0.53
French	0.54	0.55	0.57	0.54
Geology	0.55	0.56	0.5	0.56
History	0.59	0.51	0.5	0.55
Math	0.54	0.51	0.5	0.51
Quran	0.5	0.56	0.5	0.59
Science	0.57	0.55	0.52	0.56
Mathematics Course	0.52	0.5	0.57	0.57
Portuguese language Course	0.57	0.56	0.59	0.59

4.3.3 Performance metrics

In our experiments, we evaluate the clustering accuracy of L&F algorithm based on F-score, which is bound by the interval [0, 1]. It represents the harmonic mean between

the precision and recall of the clustering for all classes presented in Equation 4.1:

$$F_{combined} = \sum_{i} \frac{n_i}{n} max_j F(i,j) \tag{4.1}$$

Where:

- n: the total number of objects being clustered.
- ni : the number of objects in class i.
- max F(i,j): the maximum F-score that each class sees over all clusters given by:

$$F(i,j) = \frac{2 * Precision(i,j) * Recall(i,j)}{Precision(i,j) + Recall(i,j)} \tag{4.2}$$

Where:

- Precision (i,j): the proportion of items in cluster j that are of class i given by Equation 4.3.
- Recall (i,j): the proportion of class i that belongs to that cluster j given by Equation 4.4.

$$Precision(i,j) = \frac{n_{ij}}{n_j} \tag{4.3}$$

$$Recall(i,j) = \frac{n_{ij}}{n_i} \tag{4.4}$$

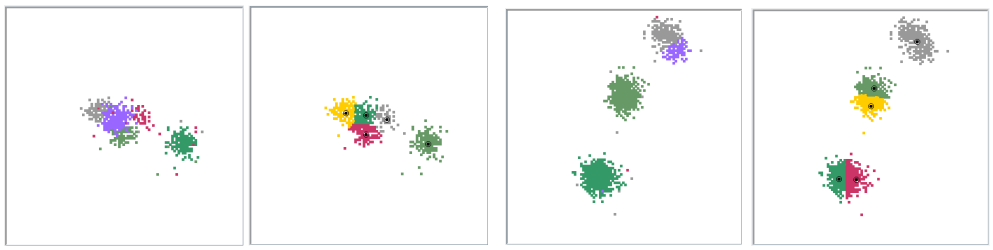
Where:

- N_{ij}: the number of objects of class i within cluster j.

4.4 Hybrid ant based clustering algorithm: Evaluation results and discussion

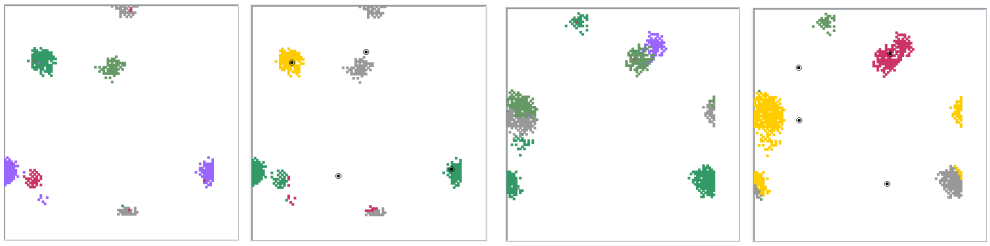
To evaluate the student group formation generated by the proposed bio-inspired KM-AC method, we performed 100,000 successive iterations to determine the convergence (stagnation) measures for cluster formation. The following subsections present these measures along with their interpretations. We then discuss the results related to performance, similarity degree, and clustering quality of the algorithms in order to identify the most effective solution.

4.4.1 Clusters Outcomes Visualization across KM-AC method



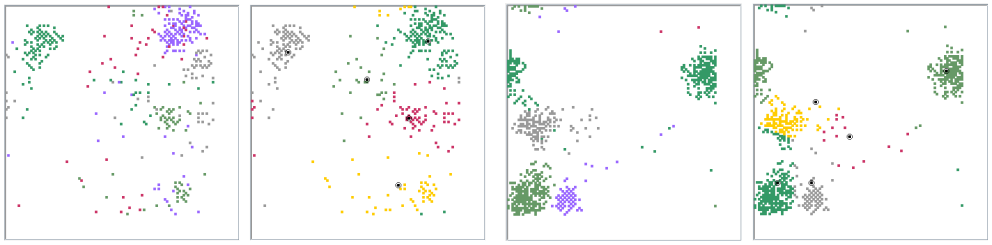
(a) L&F Data Visualization of Mathematics subject data set with and without applying Kmeans

(b) L&F Data Visualization of Language subject data set with and without applying Kmeans



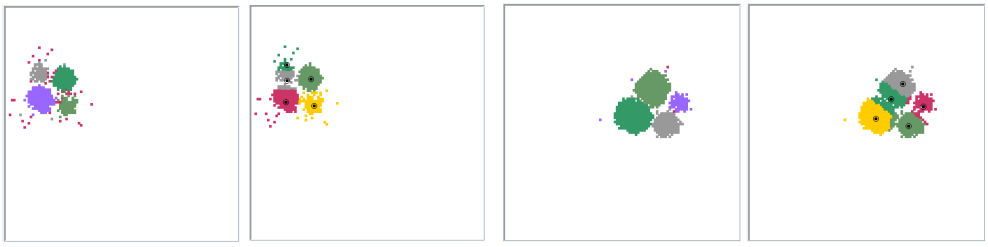
(c) ACA Data Visualization of Mathematics subject data set with and without applying Kmeans

(d) ACA Data Visualization of Language subject data set with and without applying Kmeans



(e) ACAM Data Visualization of Mathematics subject data set with and without applying Kmeans

(f) ACAM Data Visualization of Language subject data set with and without applying Kmeans



(g) Improved ACA Data Visualization of Mathematics subject data set with and without applying Kmeans

(h) Improved ACA Data Visualization of Language subject data set with and without applying Kmeans

Figure 4.4: Data visualization with and without Kmeans

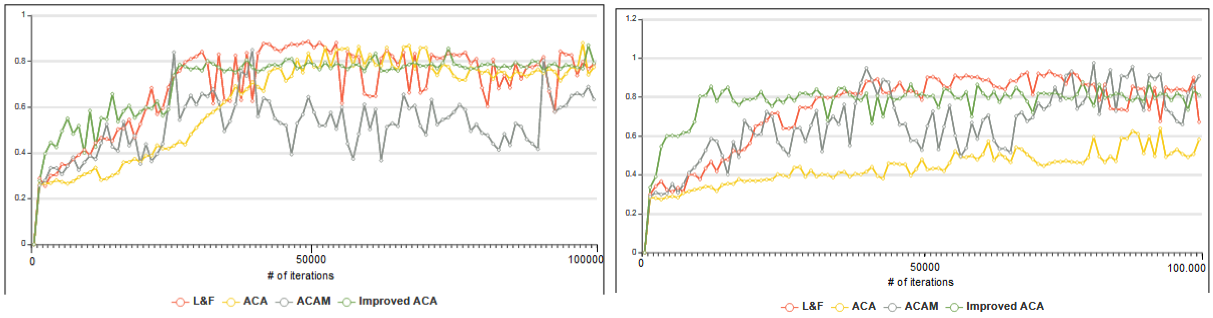
Figure 4.4 presents the visualization of clustering results by using the four ant clustering algorithms. We used 2-dimensional space to randomly project data on the grid. During a successive 100000 iterations, the algorithms showed not equally sized clusters. In

figures 4.4a and 4.4b, we apply L&F with the two Data sets. It shows very converged groups of students with few outliers to be easy to distinguish between them, even if we don't apply the k-means for Mathematics subject data set.
In the other hand, ACA (see figures 4.4c and 4.4d) shows diverse spaced clusters, so that we can see some groups are still incompleted because their elements are gathered elsewhere in the grid. Consequently, this is another reason that let us using the k-means algorithm. In figures 4.4e, 4.4f, the ACAM shows many outliers on the grid that are often much present in the Mathematic Subject data set than the other one. This is due to the number of students who are two times larger in Language data set. This is one of the reasons mentioned above in last section that required the integration of k-means. We can also conclude according to the data visualized in the two data sets that ACAM algorithm may will show more stable clusters without free items if we applied it with more larger data sets of [900..1000] instances.
As it presented with L&F algorithm, the Improved ACA in figures 4.4g and 4.4h shows a very closed clusters with few no assigned items, this is due its parameters that need more investigation in further work. As we defined that we initialized the following parameters in table 4.1 to run and evaluate KM-AC approach. For Improved ACA algorithm, we assigned a new picking-up and dropping probabilities defined by Gao (2016) using the sigmoid function as in equation 4.5:

$$Sigmoid(x) = \frac{1 - e^{-cx}}{1 - e^{+cx}} \tag{4.5}$$

with c is a slope constant that force ant to drop the free items at the ending of simulation as reported Gao (2016). According to the experiments, a large constant c lead us to very spaced out clusters as it visualized by ACA. Contrariwise, a small constant c lead us to very converged groups with some wrong assigned items that reduce the clustering performance of the algorithm.
Hence, we can note that the improved ACA is a good performed ant-clustering but it required an automatic constant c adjustment in further future work.

4.4.2 Performance Evaluation of KM-AC



(a) (b)

Figure 4.5: F-measure Results Comparison between Ant Colony based Clustering algorithms:
(a) Mathematics subject data set Results
(b) Language subject data set Results

As illustrated in Figure 4.5, the performance of the evaluated clustering algorithms varies significantly across different educational datasets, reflecting the sensitivity of these methods to the data characteristics. Notably, the F-measure results of most algorithms fluctuate, indicating instability in their clustering performance. In contrast, the Improved ACA consistently maintains a stable final value throughout the iterations, showcasing its robustness and reliability. This characteristic of stability is a significant advantage, as it suggests that the Improved ACA can deliver reliable clustering results even as the dataset evolves during the clustering process.

In Figure 4.5(a), the performance scores of the LF, ACA, and Improved ACA algorithms fall within the range of [0.7736, 0.8795] by the final iterations, with all three methods achieving relatively similar outcomes. However, a more detailed examination of the initial iterations reveals that the Improved ACA converges more quickly compared to the others. This faster convergence is crucial, particularly when dealing with large datasets where computational efficiency is important. In the first 20,000 iterations, the Improved ACA reaches values between [0.7228, 0.8749], demonstrating its ability to deliver high-quality clustering results in a shorter amount of time. This rapid initial performance makes the Improved ACA an ideal choice for applications that require efficient clustering within a limited number of iterations, such as early-stage data analysis or real-time clustering applications.

Similarly, in Figure 4.5(b), the Improved ACA continues to exhibit superior performance, maintaining a steady improvement in clustering quality throughout the iterations. It surpasses a Rand Index of 0.6 after just 104 iterations, demonstrating that it is capable

of providing high-quality clusters even in the early stages of the process. Interestingly, the Improved ACA performs better with the larger dataset containing 649 students than with the smaller dataset of 395 students, indicating that the algorithm's performance benefits from a larger volume of data. This suggests that the Improved ACA is well-suited for handling more complex datasets, where the increased number of instances may help the algorithm achieve more distinct and meaningful clusters. In contrast, the ACA algorithm shows weaker performance with the Language subject dataset, further reinforcing the idea that the algorithm's effectiveness is influenced by the size and characteristics of the dataset.

This variability in performance highlights the importance of choosing the right algorithm for the right dataset. The ACA, in particular, seems to depend on the number of instances in the dataset to achieve good clustering results. As the number of instances increases, the ACA's performance becomes less stable, especially when dealing with more complex datasets. On the other hand, the LF algorithm demonstrated more stable performance with the Language subject dataset, suggesting that it might be more robust in certain contexts, though it still lags behind the Improved ACA in terms of speed and overall effectiveness.

Lastly, the ACAM algorithm's performance was generally unsatisfactory across all datasets. The results suggest that ACAM lacks the stability and reliability necessary to perform accurate clustering, especially when dealing with heterogeneous or larger datasets. This instability makes it a less viable option for grouping students, as it may fail to consistently produce meaningful clusters. In contrast, the Improved ACA algorithm stands out as the most reliable and efficient choice, particularly for large datasets with a significant number of instances.

4.4.3 Similarity Degrees of ACC Algorithms with K-means

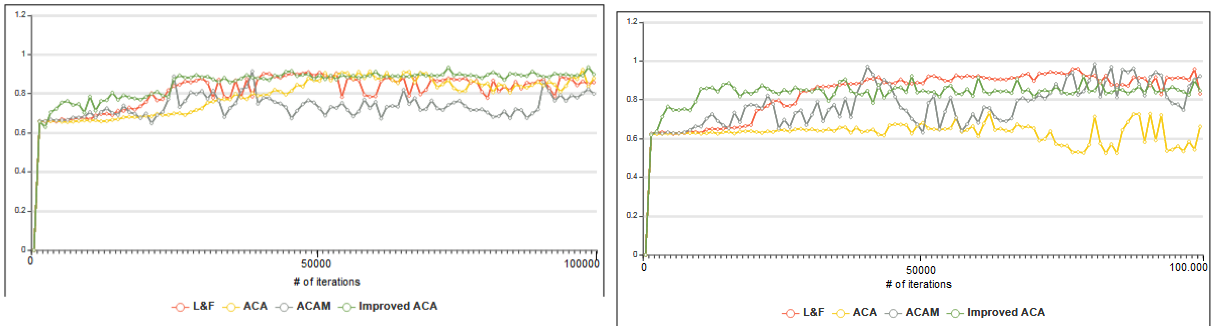
In general, better clustering performance is directly correlated with higher similarity degrees, which are assessed using various evaluation metrics. The Rand Index is one of the most widely used metrics for evaluating clustering quality, as it measures the degree of similarity between two partitions. It ranges from 0 to 1, where a value of 1 indicates a perfect match, meaning that the clusters from one partition are identical to those from the other. A higher Rand Index value indicates that the algorithm has produced more accurate and consistent clusters. The results shown in Figure 4.6 strongly suggest that the Improved ACA algorithm is highly effective in grouping students in datasets (a) and (b), achieving better clustering results than the other tested algorithms.

In dataset (a), the Improved ACA exhibits rapid improvements in similarity degrees

early on, with a marked increase in clustering quality. It also maintains a higher level of stability throughout the iterations, outperforming the other algorithms in terms of consistency. This stability is crucial, as it indicates that the Improved ACA is not only able to identify high-quality clusters but can do so reliably, without fluctuations that could lead to inconsistent results. On the other hand, in dataset (b), although the Improved ACA still demonstrates reasonable performance, it does not maintain the same level of stability observed in dataset (a). This suggests that the algorithm's effectiveness can be affected by the dataset characteristics, such as size or complexity. In this case, the LF algorithm performs better in terms of stability, providing a more consistent Rand Index measure throughout the iterations.

The performance of the ACAM algorithm stands in contrast to that of the Improved ACA. In dataset (a), ACAM shows the lowest similarity degree evolution, struggling to improve its clustering performance in the early stages of the process. While it does manage to achieve satisfactory similarity measures toward the end of the clustering process, its slow convergence and lower overall similarity scores limit its effectiveness. This suggests that ACAM might not be the best option when rapid convergence and early stability are required for clustering tasks. The ACA algorithm, although showing promising Rand Index results with the Mathematics subject dataset, exhibits a significant drop in performance when applied to larger datasets, such as the Portuguese language dataset. Here, the similarity measures are substantially lower, highlighting that the ACA's ability to produce effective clusters is heavily influenced by dataset size. This inconsistency further suggests that ACA may struggle with more complex datasets, where the relationships between data points are less clear or more varied.

These findings underscore the critical role that dataset size and complexity play in the performance of clustering algorithms. The ability of an algorithm to achieve optimal Rand Index values is not only determined by the quality of the method itself but also by the nature of the dataset it is applied to. Smaller, simpler datasets may allow algorithms to achieve higher clustering accuracy, while larger, more complex datasets present additional challenges. Therefore, selecting the most appropriate clustering algorithm requires a careful evaluation of the dataset's characteristics—such as its size, feature distribution, and the inherent structure of the data—as well as the specific clustering objectives. These factors must be taken into account to ensure that the algorithm chosen will produce the best possible results.



(a) (b)

Figure 4.6: Rand Index Results Comparison between Ant Colony based Clustering algorithms:
(a) Mathematics subject data set Results
(b) Language subject data set Results

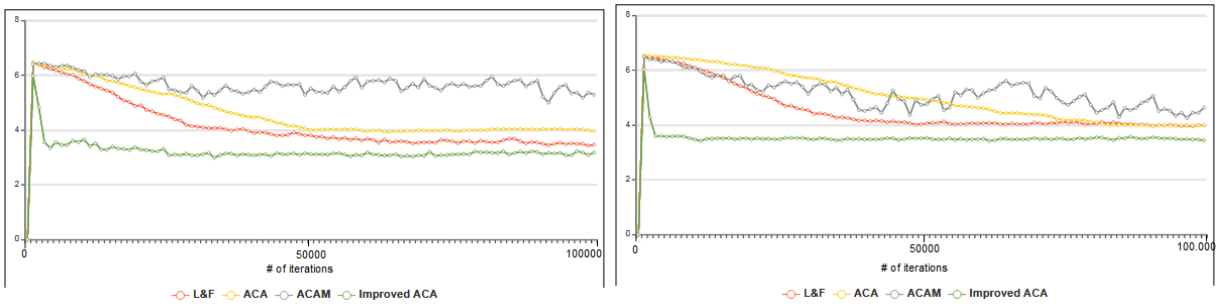
4.4.4 Evaluation of the Algorithms' Clustering Quality

The Improved ACA algorithm demonstrated the most satisfactory clustering performance and similarity degree measurements among all the methods tested. As shown in Figure 4.7, this algorithm consistently achieved the lowest entropy value, a crucial metric for evaluating the quality of clusters. Entropy, which measures the level of disorder or impurity within clusters, reflects how well the algorithm is able to group similar data points together. A lower entropy value signifies that the algorithm has formed more homogeneous and distinct clusters, indicating superior clustering performance. This result highlights the effectiveness of the Improved ACA, particularly in handling larger, more complex datasets, where the ability to identify and maintain pure clusters is crucial.

These findings strongly suggest that the Improved ACA provides an optimal solution for clustering tasks, especially in scenarios involving large and diverse datasets, such as categorizing students based on multiple attributes. The algorithm's ability to maintain high clustering quality, even as the dataset grows in size and complexity, positions it as a robust method for data grouping and pattern recognition in real-world applications. Its effectiveness is not limited to simple datasets but extends to more intricate ones, where it can identify meaningful relationships and groupings within the data more reliably than many other algorithms.

Given the promising performance demonstrated in this study, further investigation into the accuracy and scalability of the Improved ACA is necessary. Exploring its behavior with even larger and more complex datasets, as well as examining its performance across a broader range of domains, will be essential to fully understand its capabilities. Future research should focus on fine-tuning the algorithm to handle extreme variations in

data characteristics, ensuring that its advantages are preserved across diverse use cases. This could include exploring its application in areas such as large-scale customer segmentation, social network analysis, or even medical data clustering. By expanding the scope of its applications, we can further refine the algorithm and establish the Improved ACA as a go-to tool for large-scale data analysis, capable of handling both structured and unstructured data with high efficiency and accuracy.



(a) (b)

Figure 4.7: Entropy measure Results Comparison between Ant Colony based Clustering algorithms:
(a) Mathematics subject data set Results
(b) Language subject data set Results

4.4.5 KM-AC with other Clustering Algorihtms

In this section, we present a comprehensive comparison of our hybrid method with several well-established clustering algorithms to assess its effectiveness. To carry out this evaluation, we utilized WEKA, a powerful and widely used tool for machine learning algorithm testing and performance assessment. WEKA offers multiple cluster evaluation modes, each suited to different types of analysis. For our study, we selected the Classes to Clusters evaluation mode, which is particularly useful in the context of supervised clustering tasks.

In the Classes to Clusters mode, the class attribute is initially disregarded during the clustering process. This ensures that the clustering algorithm focuses solely on the inherent structure and patterns within the data, without any bias introduced by predefined class labels. During the testing phase, each cluster is assigned a class label based on the majority class value of the instances within that cluster. Subsequently, the classification error is computed based on these assignments, and a corresponding confusion matrix is generated to visualize the clustering accuracy and misclassification rates. This process allows for an objective comparison of clustering performance across different

algorithms, with a particular focus on how well the algorithms can group instances of similar classes together.

As summarized in Table 4.3, the results of our experiments show that our hybrid method consistently outperformed the other tested algorithms across various scenarios. In particular, the ACAM algorithm demonstrated the lowest classification error for smaller datasets, indicating its efficiency in handling simpler and less complex data. However, when applied to larger datasets, the Improved ACA algorithm exhibited superior performance, achieving the least classification error and providing the most stable results. This performance disparity highlights the strength of the Improved ACA, which is better suited for large, complex datasets, where it can more effectively capture underlying patterns and relationships.

These findings underscore the adaptability and efficiency of our hybrid approach. The ACAM algorithm is highly effective for small datasets, while the Improved ACA proves to be a robust solution for larger datasets, providing flexibility and scalability across different clustering tasks. The results confirm that our hybrid method is not only competitive with traditional clustering algorithms but also excels in diverse conditions, making it a reliable choice for clustering problems of varying scale and complexity.

Table 4.3: Comparative error between clustering algorithms

Datasets	**KM-L&F%**	**KM-ACA%**	**KM-ACAM%**	**KM-Improved ACA%**	**Simple k-means%**	**EM %**	**HC %**
History	57.89	57.89	**36.84**	52.63	**36.84**	**36.84**	**36.84**
Math	42.86	42.86	**4.76**	38.09	57.14	42.86	52.38
Quran	54.55	36.36	**31.89**	59.09	40.91	**31.81**	45.45
Geology	33.33	41.67	**29.17**	50.00	66.67	50.00	50.00
Chemistry	54.17	37.50	**20.83**	54.17	41.67	38.33	41.67
Spanish	60.00	52.00	**24.00**	52.00	40.00	48.00	56.00
Biology	56.67	36.67	**20.00**	50.00	53.33	43.33	46.67
English	26.67	35.56	**15.56**	57.78	48.89	40.00	60.00
Science	37.26	41.18	**7.84**	50.98	47.06	47.06	52.94
French	27.69	36.92	**12.31**	60.00	38.46	38.46	55.38
Arabic	45.76	33.90	**10.17**	50.85	38.98	42.37	55.93
IT	28.42	23.16	**14.74**	56.84	46.32	40.00	54.74
Mathematics	68.86	34.64	39.65	**31.65**	73.67	70.13	66.84
Portuguese	67.64	46.07	45.61	**33.74**	71.96	66.56	68.72

4.4.6 Analysis of KM-AC: Stagnation Detection

As previously mentioned, the maximum number of iterations for our algorithms was initially set to 105. However, during the execution of these algorithms, we observed that after this number of iterations, all the ants converged to a single solution, leading to a situation where the algorithms became almost stagnant. This issue, commonly referred to as the "Search Stagnation Problem," is a well-known challenge that affects all Ant Colony Optimization (ACO) algorithms, regardless of the specific application domain Aljanaby (2016). The stagnation occurs when the search process fails to explore new solutions, causing the algorithm to get stuck at a suboptimal or local solution, which hinders further improvement and prevents the discovery of better solutions.

To address this issue, we conducted an experimental analysis of the stagnation behavior exhibited by our algorithms. For this purpose, we selected the Improved ACA algorithm and applied it to one of the larger datasets available the "Portuguese language" dataset, which contains 649 instances. Figure 4.8 presents the results of tracing 10 typical runs of the Improved ACA algorithm, providing insights into its behavior over time. As the Figure shows, it becomes evident that after approximately 15,000 iterations, the algorithm begins to exhibit signs of stagnation. This is characterized by a plateau in the performance, where the solution quality no longer improves despite further iterations.

To overcome this stagnation and ensure more effective exploration of the solution space, we decided to extend the number of iterations to 20,000 for the remaining algorithms. This adjustment was made to give the algorithms more time to explore potential solutions and avoid premature convergence. By extending the iteration limit, we aimed to improve the algorithm's robustness and prevent it from getting trapped in suboptimal solutions too early in the search process. This experiment highlights the importance of carefully managing the iteration limit in Ant Colony algorithms to maintain their exploratory capabilities and achieve more reliable and accurate results.

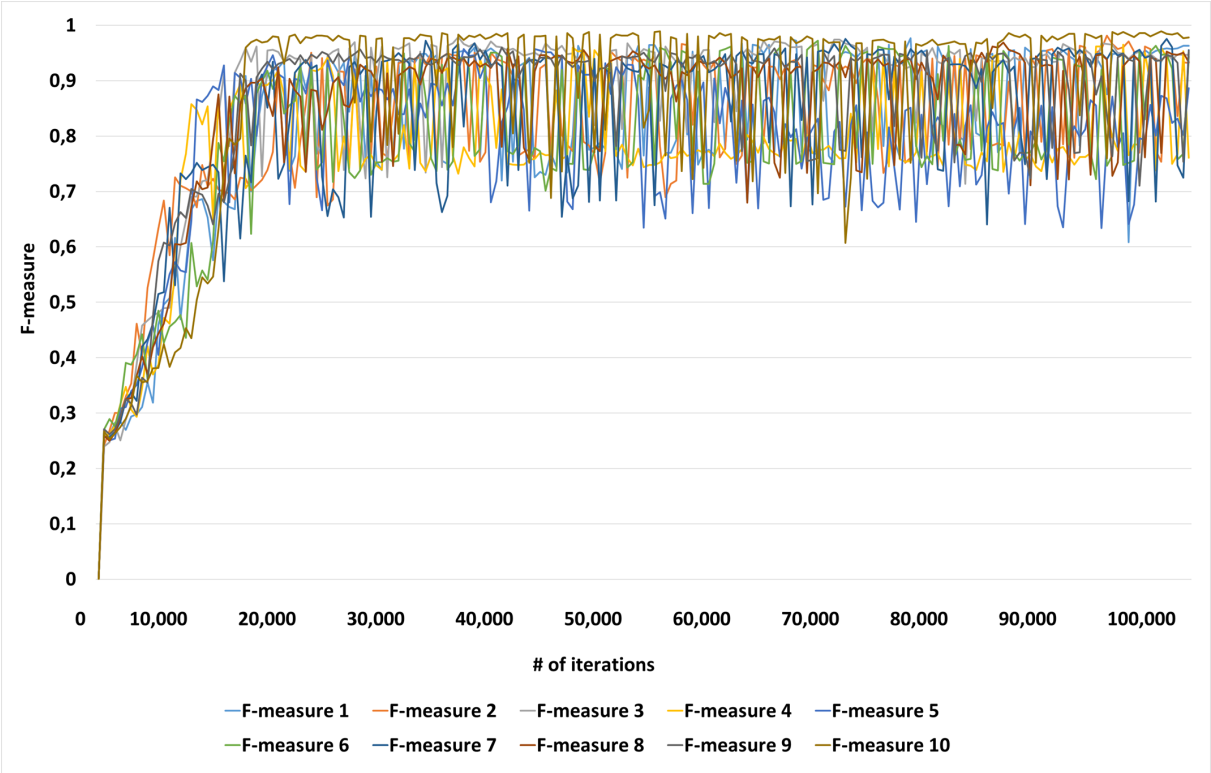


Figure 4.8: Stagnation detection by running Improved ACA algorithm 10 times using portoguse dataset

F-measure results based on KM-AC Stagnation

To provide a comprehensive comparison of our selected Ant algorithms, we conducted an experimental investigation into their performance within educational domains. The results, as presented in Figure 4.9 and further detailed in Table 4.4, summarize the end-run means for 20,000 iterations across different datasets. These findings offer valuable insights into the strengths and limitations of each algorithm in the context of clustering educational data.

From these results, two key conclusions can be drawn. First, the Improved ACA algorithm consistently outperforms the other algorithms when applied to both the Mathematics and Language datasets. This indicates that the Improved ACA excels in handling larger and more complex datasets, where the increased number of instances provides the algorithm with more opportunities to explore and identify meaningful patterns. Its superior performance suggests that the Improved ACA is particularly well-suited for tasks involving large-scale educational datasets, where scalability and accuracy are critical for effective clustering.

Second, the ACAM algorithm shows a distinct advantage when applied to smaller datasets, outperforming the other tested algorithms in this scenario. This suggests that

ACAM is more efficient at handling datasets with a reduced number of instances, where the complexity is lower and the clustering task may involve less variability in the data. The ability of ACAM to perform well with smaller datasets demonstrates its suitability for applications where data is more limited or focused, and where rapid convergence and simplicity are valued over handling large amounts of data.

These experimental results highlight the importance of selecting the appropriate algorithm based on the size and complexity of the dataset. While the Improved ACA is the preferred choice for larger datasets due to its robustness and scalability, ACAM proves to be effective for smaller, more focused datasets, where its efficiency in managing fewer instances provides an advantage. Overall, these findings emphasize the need for a tailored approach to clustering in educational fields, ensuring that the chosen algorithm aligns with the specific characteristics and requirements of the data at hand.

Table 4.4: Comparative F-score results of KM-L&F with KM-ACA, KM-ACAM and KM-Improved ACA

Datasets	**KM-L&F**	**KM-ACA**	**KM-ACAM**	**KM-Improved ACA**
History	0.421	0.530	**0.825**	0.457
Biology	0.5	0.418	**0.835**	0.451
Arabic	0.605	0.651	**0.864**	0.382
IT	0.688	0.807	**0.832**	0.811
Mathematics Course	0.426	0.354	0.524	**0.786**
Portuguese language Course	0.391	0.322	0.414	**0.830**

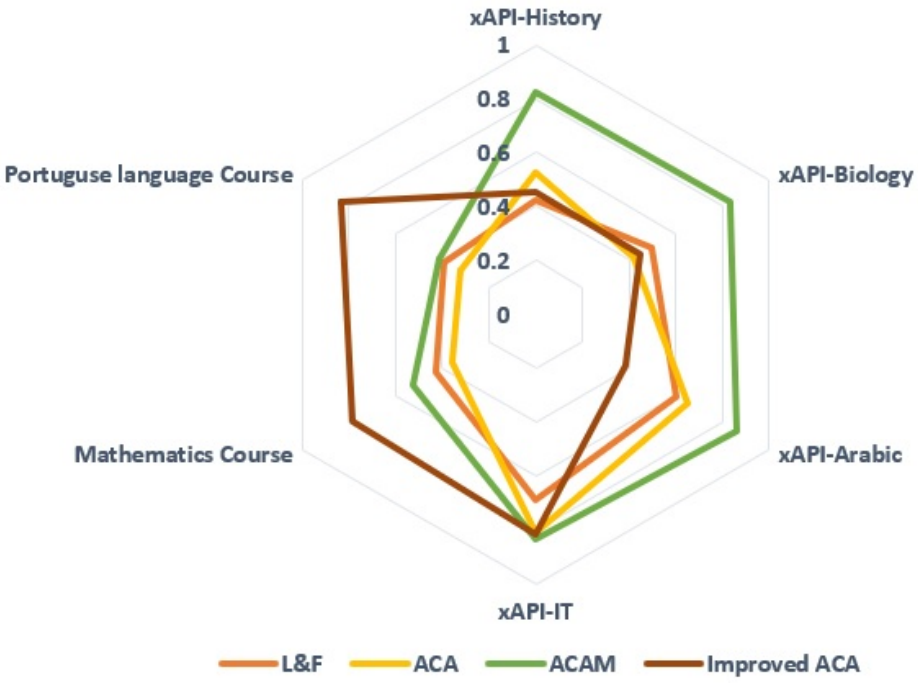


Figure 4.9: F-measure results after stagnation detection

Entropy and RandIndex for larger datasets

To further support our previous experiments investigating the reliability of the Improved ACA algorithm with larger datasets, and to provide a deeper analysis of its accuracy, this section presents the Entropy and Rand Index results for the Improved ACA across all datasets. These metrics are crucial for evaluating the quality and stability of the clustering results, as they offer valuable insights into both the consistency of the clustering process and the degree of agreement between the obtained clusters and the true class labels.

As illustrated in Figure 4.10 and Figure 4.11, the Rand Index and Entropy results for the Improved ACA algorithm exhibit a notably higher level of stability when applied to larger datasets compared to the other algorithms in the comparison. The Rand Index, which ranges from 0 to 1, measures the degree of similarity between two clustering results, with a value of 1 indicating a perfect match. The Entropy metric, on the other hand, evaluates the level of disorder within the clusters, with lower values indicating better clustering performance. In both metrics, the Improved ACA maintains more consistent performance across larger datasets, demonstrating its robustness and effectiveness in handling complex data. This stability suggests that the algorithm is better at managing the increased complexity and variability that comes with larger datasets, making it more reliable for large-scale clustering tasks.

In contrast, the other compared algorithms showed more variability in their Entropy and Rand Index values, especially as the dataset size increased. This further emphasizes the Improved ACA's superior ability to maintain high-quality clustering, even as the data grows in size and complexity. These results strengthen the argument for the Improved ACA as an optimal choice for clustering large and diverse datasets, where both accuracy and stability are paramount.

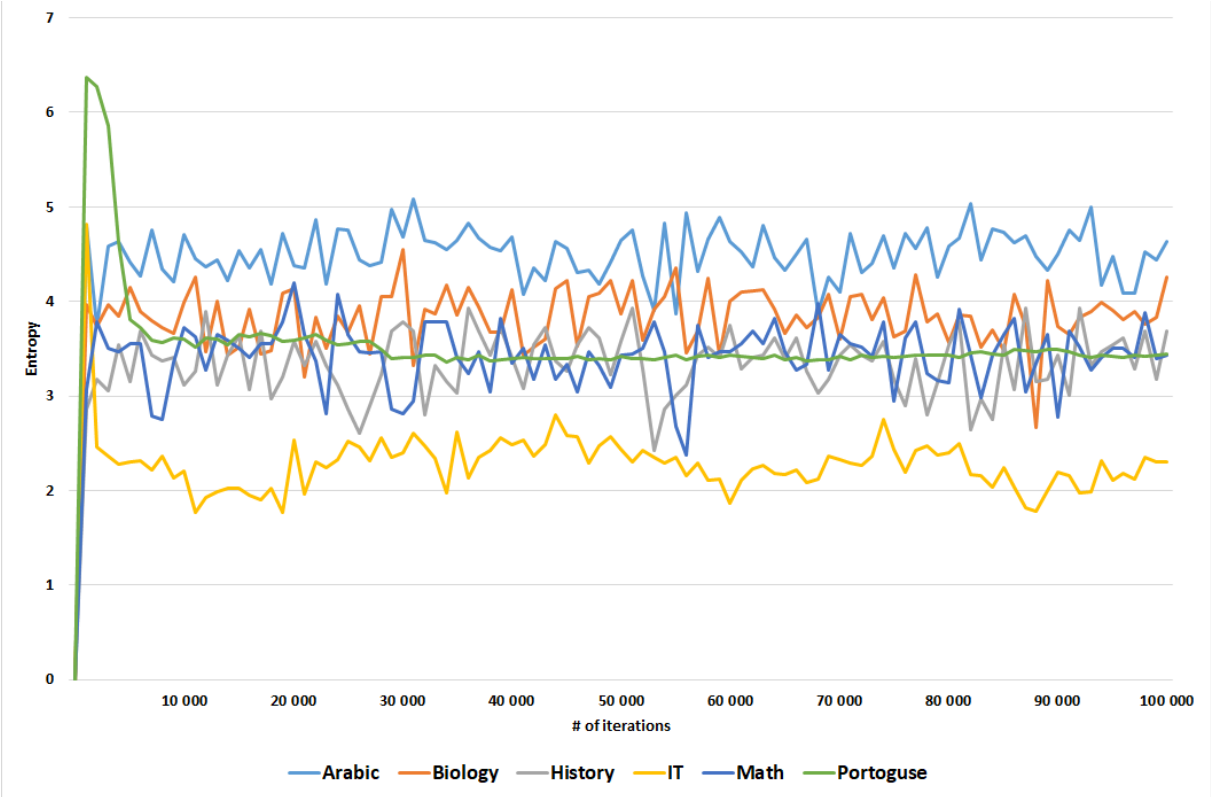


Figure 4.10: Entropy for all algorithms

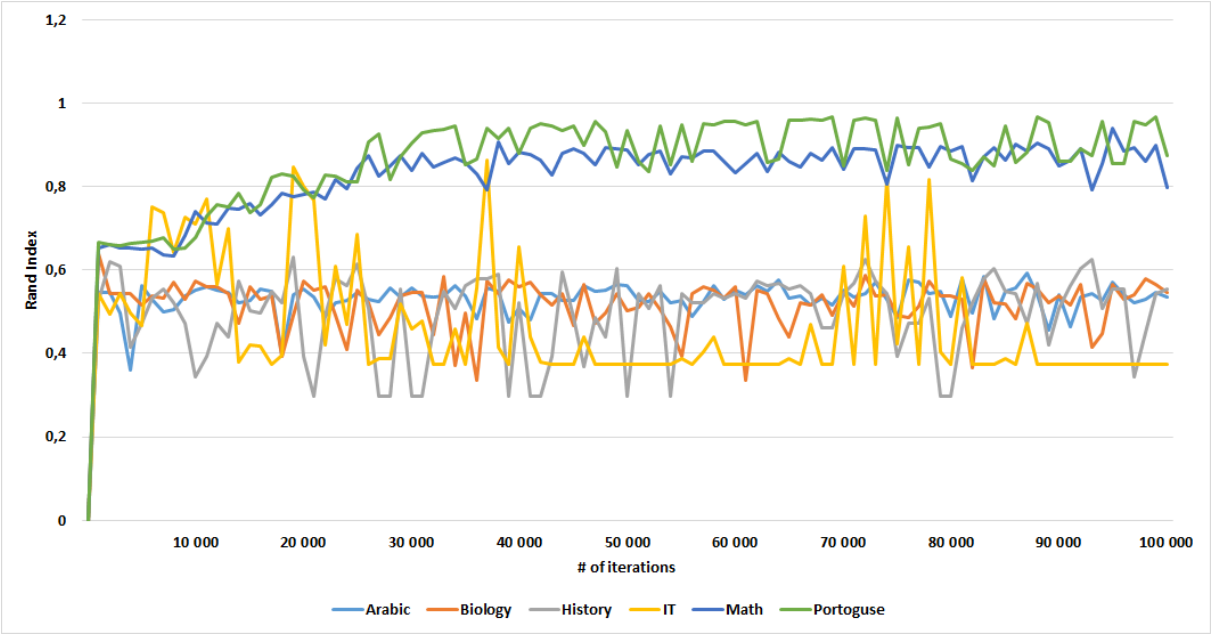


Figure 4.11: RandIndex for all algorithms

4.5 Conclusion

Forming well-balanced learner groups is essential for fostering effective collaborative learning, as it directly influences learning experiences and the overall outcomes. In this chapter, we introduce a hybrid clustering approach that combining the exploratory capabilities of the Ant Colony-based Clustering Algorithm with the precision of the deterministic K-means algorithm. This integration was motivated by the need to address group formation without outliers, ensuring every student is assigned to a suitable group. Our findings demonstrate that the size of the dataset significantly affects clustering performance, particularly for algorithms like Ant Colony-based Clustering Algorithm sensitive to data distribution. K-means helps by counteract refining the clustering boundaries and by reinforcing the solution robustness. Overall, our hybrid model exhibits superior adaptability and scalability, consistently outperforming traditional clustering algorithms across varying data sizes.

Conclusion

Creating appropriate learner groups is fundamental to successful collaborative learning environments. This thesis systematically addressed this challenge through four complementary contributions, culminating in the hybrid KM-AC algorithm that guarantees complete partitioning of educational datasets without outliers.

Research Objectives and Contributions

Our investigation was structured around the core problem of ant colony clustering's limitations: parameter sensitivity and incomplete partitioning. Chapter 1 established the theoretical foundation through a systematic literature review of collaborative learning and four key ant colony algorithms (L&F, ACA, ACAM, Improved ACA), identifying their non-deterministic nature as the primary barrier to guaranteed student assignment.

Chapter 2 analyzed feature selection techniques, demonstrating that ReliefF feature selection consistently outperformed other methods when processing educational datasets blending academic and social attributes. This preprocessing step proved essential for meaningful clustering.

Chapter 3 provided critical insight into parameter sensitivity, revealing that ant colony performance depends heavily on the α parameter and dataset size. Improved ACA excelled on large datasets while ACAM performed better on smaller ones, establishing clear tuning guidelines.

Finally, Chapter 4 delivered the KM-AC hybrid algorithm, combining ant colony exploration with K-means deterministic precision. This fusion eliminated outliers across dataset scales, achieving 4.76% classification error on small datasets compared to 38.09% for traditional methods.

Theoretical Contributions

This work advances bio-inspired clustering theory in three significant ways:

- Dataset-specific α parameter guidelines provide practitioners with reproducible tuning strategies absent from prior literature

- ReliefF superiority for educational data establishes a new preprocessing benchmark
- KM-AC hybrid model transforms stochastic ant colony methods into deterministic solutions suitable for production educational systems

Practical Implications

Educators and e-learning platforms gain a deployable KM-AC framework for automated group formation that optimizes academic complementarity and social dynamics while ensuring every student belongs to a group. The feature selection pipeline (Chapter 2) and parameter guidelines (Chapter 3) enable immediate implementation across classroom and MOOC scales.

Limitations

While α sensitivity is well-characterized, other ant colony parameters (evaporation rate, number of ants) require similar analysis. KM-AC's computational complexity may limit real-time applications. Evaluation focused on academic/social attributes; generalization to other domains needs validation.

Future Research Directions

Promising extensions include:

- Automatic α tuning via an open source hyperparameter optimization framework to automate hyperparameter search (OPTUNA) Ozaki et al. (2025), or heuristic based machine learning system.
- Dynamic regrouping for real-time student addition/removal in collaborative teams
- Emotion-aware grouping incorporating affective computing features
- Parallel implementation for MOOC-scale deployments
- Longitudinal validation measuring actual learning outcomes

This thesis transforms ant colony clustering from exploratory research tool to production-ready solution for intelligent student grouping. By systematically addressing feature optimization, parameter sensitivity, and partitioning guarantees, KM-AC establishes bio-inspired algorithms as cornerstone technology for next-generation collaborative learning systems.

Appendix A: Additional Tables from Chapter 2

.1 The following tables provide additional data referenced in Chapter 2.

Table 1: Full Features Set of "xAPI Edu Dataset"

Feature subset	Attributes	Nb of Features
Full Features Set	StudentAbsenceDays, VisITedResources, PlaceofBirth, NationalITy, gender, Discussion, ParentAnsweringSurvey, raisedhands, Relation, SectionID, StageID, GradeID, ParentschoolSatisfaction, Semester, AnnouncementsView.	15

Table 2: Selected Attributes Using Feature Selection Techniques for "xAPI-IT Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	NationalITy, Relation, raisedhands, VisITedResources, AnnouncementsView, ParentAnsweringSurvey, StudentAbsenceDays	7
Rf Features	StudentAbsenceDays, raisedhands, VisITedResources, AnnouncementsView, ParentAnsweringSurvey, SectionID, ParentschoolSatisfaction, Relation, gender, Discussion, NationalITy, PlaceofBirth, GradeID	13
IG Features	tudentAbsenceDays, GradeID, StageID, Relation, PlaceofBirth, raisedhands, ParentschoolSatisfaction, ParentAnsweringSurvey, NationalITy, AnnouncementsView, gender, VisITedResources, SectionID	13
SU Features	tudentAbsenceDays, GradeID, StageID, Relation, PlaceofBirth, raisedhands, ParentschoolSatisfaction, ParentAnsweringSurvey, NationalITy, AnnouncementsView, gender, VisITedResources, SectionID	13

Table 3: Selected Attributes Using Feature Selection Techniques for "xAPI-Spanish Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	Relation, raisedhands, ParentAnsweringSurvey, StudentAbsenceDays	4
Rf Features	Relation, raisedhands, StudentAbsenceDays, VisITedResources, ParentAnsweringSurvey, Discussion, StageID, GradeID, AnnouncementsView	9
IG Features	raisedhands, Relation, PlaceofBirth, NationalITy, StudentAbsenceDays, ParentAnsweringSurvey, SectionID, GradeID, StageID, gender, ParentschoolSatisfaction	11
SU Features	raisedhands, Relation, PlaceofBirth, NationalITy, StudentAbsenceDays, ParentAnsweringSurvey, SectionID, GradeID, StageID, gender, ParentschoolSatisfaction	11

Table 4: Selected Attributes Using Feature Selection Techniques for "xAPI-Arabic Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	NationalITy, raisedhands, VisITedResources, ParentAnsweringSurvey, StudentAbsenceDays	5
Rf Features	StudentAbsenceDays, raisedhands, VisITedResources, ParentAnsweringSurvey, ParentschoolSatisfaction, Relation, Discussion, AnnouncementsView, StageID, GradeID	10
IG Features	VisITedResources, NationalITy, raisedhands, StudentAbsenceDays, PlaceofBirth, ParentAnsweringSurvey, ParentschoolSatisfaction, Relation, GradeID, SectionID, StageID, gender, Semester	13
SU Features	VisITedResources, NationalITy, raisedhands, StudentAbsenceDays, PlaceofBirth, ParentAnsweringSurvey, ParentschoolSatisfaction, Relation, GradeID, SectionID, StageID, gender, Semester	13

Table 5: Selected Attributes Using Feature Selection Techniques for "xAPI-Biology Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	raisedhands, VisITedResources, ParentschoolSatisfaction, StudentAbsenceDays	4
Rf Features	StudentAbsenceDays, VisITedResources, raisedhands, Discussion, ParentschoolSatisfaction, PlaceofBirth, ParentAnsweringSurvey, AnnouncementsView, Relation, NationalITy	10
IG Features	VisITedResources, PlaceofBirth, raisedhands, Discussion, StudentAbsenceDays, NationalITy, ParentschoolSatisfaction, Relation, ParentAnsweringSurvey, gender, SectionID	11
SU Features	VisITedResources, PlaceofBirth, raisedhands, Discussion, StudentAbsenceDays, NationalITy, ParentschoolSatisfaction, Relation, ParentAnsweringSurvey, gender, SectionID	11

Table 6: Selected Attributes Using Feature Selection Techniques for "xAPI-French Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	PlaceofBirth, VisITedResources, StudentAbsenceDays	3
Rf Features	StudentAbsenceDays, VisITedResources, gender, ParentAnsweringSurvey, Relation, NationalITy, raisedhands, ParentschoolSatisfaction, PlaceofBirth, AnnouncementsView, Discussion, StageID, GradeID	13
IG Features	StudentAbsenceDays, VisITedResources, PlaceofBirth, NationalITy, gender, Discussion, ParentAnsweringSurvey, raisedhands, Relation, SectionID, StageID, GradeID, ParentschoolSatisfaction, Semester	14
SU Features	StudentAbsenceDays, VisITedResources, PlaceofBirth, NationalITy, gender, Discussion, ParentAnsweringSurvey, raisedhands, Relation, SectionID, StageID, GradeID, ParentschoolSatisfaction, Semester	14

Table 7: Selected Attributes Using Feature Selection Techniques for "xAPI-Geology Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	gender, Relation, VisITedResources, StudentAbsenceDays	4
Rf Features	Gender, Discussion, AnnouncementsView, ParentAnsweringSurvey, ParentschoolSatisfaction, raisedhands, VisITedResources, PlaceofBirth, StageID, StudentAbsenceDays	10
IG Features	visITedResources, StudentAbsenceDays, PlaceofBirth, NationalITy, gender, Relation, ParentAnsweringSurvey, ParentschoolSatisfaction	8
SU Features	visITedResources, StudentAbsenceDays, PlaceofBirth, NationalITy, gender, Relation, ParentAnsweringSurvey, ParentschoolSatisfaction	8

Table 8: Selected Attributes Using Feature Selection Techniques for "xAPI-History Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	Relation, VisITedResources, AnnouncementsView, ParentAnsweringSurvey, ParentschoolSatisfaction, StudentAbsenceDays	6
Rf Features	ParentschoolSatisfaction, ParentAnsweringSurvey, VisITedResources, AnnouncementsView, StudentAbsenceDays, raisedhands, Relation, Discussion	8
IG Features	AnnouncementsView, VisITedResources, ParentschoolSatisfaction, ParentAnsweringSurvey, StudentAbsenceDays, Relation, NationalITy, PlaceofBirth, gender, StageID, GradeID ,Semester	12

Table 9: Selected Attributes Using Feature Selection Techniques for "xAPI-Math Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	GradeID, Relation, AnnouncementsView	3
Rf Features	VisITedResources, AnnouncementsView, Discussion, raisedhands, Relation, Semester, StageID, SectionID, gender, GradeID, StudentAbsenceDays	11
IG Features	AnnouncementsView, GradeID, StageID, NationalITy, Relation, Semester, gender, PlaceofBirth, SectionID, StudentAbsenceDays, ParentAnsweringSurvey, ParentschoolSatisfaction	12
SU Features	AnnouncementsView, GradeID, StageID, NationalITy, Relation, Semester, gender, PlaceofBirth, SectionID, StudentAbsenceDays, ParentAnsweringSurvey, ParentschoolSatisfaction	12

Table 10: Selected Attributes Using Feature Selection Techniques for "xAPI-Quran Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	PlaceofBirth, raisedhands, VisITedResources, StudentAbsenceDays	4
Rf Features	StudentAbsenceDays, raisedhands, VisITedResources, ParentAnsweringSurvey, Discussion, Relation, AnnouncementsView, ParentschoolSatisfaction, PlaceofBirth, NationalITy, gender, Semester	12
IG Features	PlaceofBirth, StudentAbsenceDays, VisITedResources, NationalITy, raisedhands, ParentAnsweringSurvey, Relation, ParentschoolSatisfaction, Semester, gender, GradeID, StageID	12
SU Features	PlaceofBirth, StudentAbsenceDays, VisITedResources, NationalITy, raisedhands, ParentAnsweringSurvey, Relation, ParentschoolSatisfaction, Semester, gender, GradeID, StageID	12

Table 11: Selected Attributes Using Feature Selection Techniques for "xAPI-Science Dataset"

Feature subset	Attributes	Nb of Features
Cfs Features	Relation, VisITedResources, AnnouncementsView, Discussion, StudentAbsenceDays	5
Rf Features	StudentAbsenceDays, ParentschoolSatisfaction, Relation, VisITedResources, ParentAnsweringSurvey, AnnouncementsView, Discussion, gender, raisedhands, GradeID, StageID	11
IG Features	AnnouncementsView, VisITedResources, StudentAbsenceDays, Discussion, raisedhands, Relation,ParentschoolSatisfaction,ParentAnsweringSurvey, PlaceofBirth, gender, GradeID, NationalITy, StageID, SectionID, Semester	15
SU Features	AnnouncementsView, VisITedResources, StudentAbsenceDays, Discussion, raisedhands, Relation,ParentschoolSatisfaction,ParentAnsweringSurvey, PlaceofBirth, gender, GradeID, NationalITy, StageID, SectionID, Semester	15

Bibliography

Abid, A., Kallel, I., and Ayed, M. B. (2016). Teamwork construction in E-learning system: A systematic literature review. In *15th International Conference on Information Technology Based Higher Education and Training, ITHET 2016, Istanbul, Turkey, September 8-10, 2016*, pages 1–7. IEEE.

Abid, A., Kallel, I., Blanco, I. J., and Ayed, M. B. (2017). Selecting Relevant Educational Attributes for Predicting Students' Academic Performance. In Abraham, A., Muhuri, P. K., Muda, A. K., and Gandhi, N., editors, *Intelligent Systems Design and Applications - 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) Held in Delhi, India, December 14-16, 2017*, volume 736 of *Advances in Intelligent Systems and Computing*, pages 650–660. Springer.

Abid, A., Kallel, I., Sanchez-Medina, J. J., and Ayed, M. B. (2023). Parameters sensitivity analysis of ant colony based clustering: Application for student grouping in collaborative learning environment. *IEEE Access*.

Abid, A., Kallel, I., Sanchez-Medina, J. J., and Ayed, M. B. (2025). Improving ant clustering algorithms through supervised method: Application in student grouping with various real datasets. *Evolutionary Intelligence*.

Abid, A., Somai, M., Kammoun, H. M., and Kallel, I. (2024). The najeh effect: How chatgpt is shaping the future of higher education. In *2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–8. IEEE.

Alharbi, S. M., Elfeky, A. I., and Ahmed, E. S. (2022). The effect of e-collaborative learning environment on development of critical thinking and higher order thinking skills. *Journal of Positive School Psychology*, pages 6848–6854.

Aljanaby, A. (2016). An experimental study of the search stagnation in ants algorithms. *International Journal of Computer Applications*, 148(14).

Amrieh, E. A., Hamtini, T., and Aljarah, I. (2015). Preprocessing and analyzing educational data set using X-API for improving student's performance. In *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–5, Amman, Jordan. IEEE.

Amrieh, E. A., Hamtini, T., and Aljarah, I. (2016). Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. *International Journal of Database Theory and Application*, 9(8):119–136.

Anari, B., Torkestani, J. A., and Rahmani, A. M. (2018). A learning automata-based clustering algorithm using ant swarm intelligence. *Expert Systems*.

Asad, M. M. and Qureshi, A. (2025). Impact of technology-based collaborative learning on students' competency-based education: insights from the higher education institution of pakistan. *Higher Education, Skills and Work-Based Learning*.

Badoni, R. P., Kumar, S., Mann, M., Mohanty, R., and Sarangi, A. (2023). Ant colony optimization algorithm for the university course timetabling problem using events based on groupings of students. In *Modeling and Applications in Operations Research*, pages 1–36. CRC Press.

Baltierra, S., Valdebenito, J., and Mora, M. (2022). A proposal of edge detection in images with multiplicative noise using the ant colony system algorithm. *Engineering Applications of Artificial Intelligence*, 110:104715.

Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519.

Bonabeau, E., Dorigo, M., Marco, D. d. R. D. F., Theraulaz, G., Théraulaz, G., et al. (1999). *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press.

Boryczka, U. (2008). Ant clustering algorithm. *Intelligent information systems*, 1998:455–458.

Boryczka, U. (2009). Finding groups in data: Cluster analysis with ants. *Applied Soft Computing*, 9(1):61–70.

Bourkoukou, O., El Bachari, E., and El Boustani, A. (2019). Building effective collaborative groups in e-learning environment. In *International Conference on Advanced Intelligent Systems for Sustainable Development*, pages 107–117. Springer.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Carlisle, A. and Dozier, G. (2001). An off-the-shelf pso. proceedings of the workshop on particle swarm optimization. *Purdue school of engineering and technology, Indianapolis, IN*.

Chatty, A., Gaussier, P., Kallel, I., Laroque, P., and Alimi, A. M. (2012). Adaptation capability of cognitive map improves behaviors of social robots. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6. IEEE.

Chatty, A., Gaussier, P., Kallel, I., Laroque, P., and Alimi, A. M. (2013). Learning by imitation for the improvement of the individual and the social behaviors of self-organized autonomous agents. In *International Conference in Swarm Intelligence*, pages 44–52. Springer.

Chatty, A., Kallel, I., Gaussier, P., and Alimi, A. M. (2011). Emergent complex behaviors for swarm robotic systems by local rules. In *Robotic Intelligence In Informationally Structured Space (RiiSS), 2011 IEEE Workshop on*, pages 69–76. IEEE.

Chavarría-Molina, J., Fallas-Monge, J. J., and Trejos-Zelaya, J. (2020). Clustering via ant colonies: Parameter analysis and improvement of the algorithm. *Advanced Studies in Behaviormetrics and Data Science: Essays in Honor of Akinori Okada*, pages 265–282.

Chen, C.-M. and Kuo, C.-H. (2019). An optimized group formation scheme to promote collaborative problem-based learning. *Computers & Education*, 133:94–115.

Chen, J. and Li, Z. (2024). Adaptive condensed fuzzy monotonic k-nearest neighbors for monotonic classification. *International Journal of Machine Learning and Cybernetics*, pages 1–20.

Cheng, F.-F., Wu, C.-S., and Su, P.-C. (2021). The impact of collaborative learning and personality on satisfaction in innovative teaching context. *Frontiers in Psychology*, 12:713497.

Chniter, M., Abid, A., and Kallel, I. (2018). Towards a Bio-inspired ACO Approach for Building Collaborative Learning Teams. In *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–8.

Chniter, M., Abid, A., Kallel, I., and Kanoun, S. (2022). Computational complexity analysis of ant colony clustering algorithms: Application to student's grouping problem. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 736–741. IEEE.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123.

Cortez, P. and Silva, A. (2008a). Using data mining to predict secondary school student performance. *EUROSIS*.

Cortez, P. and Silva, A. M. G. (2008b). Using data mining to predict secondary school student performance.

Costa, E. B., Fonseca, B., Santana, M. A., de Araújo, F. F., and Rego, J. (2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, 73:247–256.

Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1992). The dynamics of collective sorting: Robot-like ants and ant-like robots. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 353–363.

Dorigo, M., Birattari, M., and Stützle, T. (2006). Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique. *IEEE Comput. Intell. Mag*, 1:28–39.

Dorigo, M. and Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, volume 2, pages 1470–1477. IEEE.

EL MEZOUARY, A., HMEDNA, B., and Omar, B. (2019). An evaluation of learner clustering based on learning styles in mooc course. In *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, pages 1–5. IEEE.

Figueiredo, E., Macedo, M., Siqueira, H. V., Santana Jr, C. J., Gokhale, A., and Bastos-Filho, C. J. (2019). Swarm intelligence for clustering—a systematic review with

new perspectives on data mining. *Engineering Applications of Artificial Intelligence*, 82:313–329.

Forrest, S. (1996). Genetic algorithms. *ACM computing surveys (CSUR)*, 28(1):77–80.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

Gao, W. (2016). Improved Ant Colony Clustering Algorithm and Its Performance Study. *Computational Intelligence and Neuroscience*, 2016:1–14.

Gu, Q., Cai, Z., Zhu, L., and Huang, B. (2008). Data mining on imbalanced data sets. In *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on*, pages 1020–1024. IEEE.

Gyimah, E. and Dake, D. K. (2019). Using decision tree classification algorithm to predict learner typologies for project-based learning. In *2019 International Conference on Computing, Computational Modelling and Applications (ICCMA)*, pages 130–1304. IEEE.

Hall, M. A. (1999). Correlation based feature selection for machine learning.

Handl, J., Knowles, J., and Dorigo, M. (2003). Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and id-som. In *Proceedings of the Third International Conference on Hybrid Intelligent Systems, IOS Press*.

Hassan, H., Ahmad, N. B., and Anuar, S. (2020). Improved students' performance prediction for multi-class imbalanced problems using hybrid and ensemble approach in educational data mining. In *Journal of Physics: Conference Series*, volume 1529, page 052041. IOP Publishing.

Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., and Prasath, V. S. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*, 10(12):390.

Havard, B., Du, J., and Xu, J. (2008). Online collaborative learning and communication media. *Journal of interactive learning research*, 19(1):37–50.

Huang, H.-W., Wu, C.-W., and Chen, N.-S. (2012). The effectiveness of using procedural scaffoldings in a paper-plus-smartphone collaborative learning context. *Computers & Education*, 59(2):250–259.

Johnson, D. W. and Johnson, R. T. (2009). An educational psychology success story: Social interdependence theory and cooperative learning. *Educational researcher*, 38(5):365–379.

Joseph, N., Pradeesh, N., Chatterjee, S., and Bijlani, K. (2017a). A novel approach for group formation in collaborative learning using learner preferences. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1564–1568, Udupi. IEEE.

Joseph, N., Pradeesh, N., Chatterjee, S., and Bijlani, K. (2017b). A novel approach for group formation in collaborative learning using learner preferences. In *2017 international conference on advances in computing, communications and informatics (ICACCI)*, pages 1564–1568. IEEE.

Kallel, I., Chatty, A., and Alimi, A. M. (2008). Self-organizing multirobot exploration through counter-ant algorithm. In *International Workshop on Self-Organizing Systems*, pages 133–144. Springer.

Kammoun, H. M., Kallel, I., Alimi, A. M., and Casillas, J. (2011). Improvement of the road traffic management by an ant-hierarchical fuzzy system. In *2011 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS) Proceedings*, pages 38–45. IEEE.

Kao, Y. and Li, Y. L. (2008). Ant colony recognition systems for part clustering problems. *International Journal of Production Research*, 46(15):4237–4258.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee.

Kille, T., Bates, P., and Murray, P. S. (2015). Peer observation in the online learning environment: the case of aviation higher education. In *Teaching for Learning and Learning for Teaching*, pages 79–97. Brill Sense.

Kim, M. H. (2021). Effects of collaborative learning in a virtual environment on students' academic achievement and satisfaction. *Journal of Digital Convergence*, 19(4):1–8.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256.

Kiran, M. S., Siramkaya, E., Esme, E., and Senkaya, M. N. (2022). Prediction of the number of students taking make-up examinations using artificial neural networks. *International Journal of Machine Learning and Cybernetics*, 13(1):71–81.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer.

Laal, M. and Ghodsi, S. M. (2012). Benefits of collaborative learning. *Procedia-social and behavioral sciences*, 31:486–490.

Lage, M. J., Platt, G. J., and Treglia, M. (2000). Inverting the classroom: A gateway to creating an inclusive learning environment. *The journal of economic education*, 31(1):30–43.

Lewicki, A. and Pancerz, K. (2020). Ant–based clustering for flow graph mining. *International Journal of Applied Mathematics and Computer Science*, 30(3):561–572.

Li, L. and Luo, X. (2014). Automatic Student Grouping Method for Foreign Language Learning. In *2014 10th International Conference on Semantics, Knowledge and Grids*, pages 63–66, Beijing, China. IEEE.

Lumer, E. D. and Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. pages 501–508. MIT Press.

Männistö, M., Mikkonen, K., Kuivila, H.-M., Virtanen, M., Kyngäs, H., and Kääriäinen, M. (2020). Digital collaborative learning in nursing education: a systematic review. *Scandinavian journal of caring sciences*, 34(2):280–292.

Márquez-Vera, C., Morales, C. R., and Soto, S. V. (2013). Predicting school failure and dropout by using data mining techniques. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 8(1):7–14.

Miller, A. (2002). *Subset selection in regression*. CRC Press.

Mitra, P., Murthy, C., and Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):301–312.

Nannen, V. and Eiben, A. E. (2006). A method for parameter calibration and relevance estimation in evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 183–190.

Noura, A., Shili, H., and Romdhane, L. B. (2016). Reliable attribute selection based on random forest (raser). In *International Conference on Intelligent Systems Design and Applications*, pages 11–24. Springer.

Ozaki, Y., Watanabe, S., and Yanase, T. (2025). OptunaHub: A platform for black-box optimization. *arXiv preprint arXiv:2510.02798*.

Pang, Y., Xiao, F., Wang, H., and Xue, X. (2014). A clustering-based grouping model for enhancing collaborative learning. In *2014 13th International Conference on Machine Learning and Applications*, pages 562–567. IEEE.

Pattanpichet, F. et al. (2011). The effects of using collaborative learning to enhance students english speaking achievement. *Journal of College Teaching & Learning (TLC)*, 8(11):1–10.

Pawar, P. S., Saini, J. R., Pawar, P., and Vaidya, S. (2024). Genetic algorithm application for improving the performance of teaching learning process through collaborative learning. In *Doctoral Symposium on Computational Intelligence*, pages 321–335. Springer.

Peng, C.-C., Tsai, C.-J., Chang, T.-Y., Yeh, J.-Y., and Lee, M.-C. (2020). Novel heterogeneous grouping method based on magic square. *Information Sciences*, 517:340–360.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1996). *Numerical recipes in C*, volume 2. Cambridge university press Cambridge.

Priyadarshi, R. and Kumar, R. R. (2025). Evolution of swarm intelligence: A systematic review of particle swarm and ant colony optimization approaches in modern research. *Archives of Computational Methods in Engineering*, pages 1–42.

Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Ramaswami, M. and Bhaskaran, R. (2009). A study on feature selection techniques in educational data mining. *arXiv preprint arXiv:0912.3924*.

Revelo-Sánchez, O., Ordóñez, C. A. C., and Duque, M. Á. R. (2021). Group formation in collaborative learning contexts based on personality traits: An empirical study in initial programming courses. *IxD&A*, 49:29–45.

Sahana, S. K. et al. (2019). An automated parameter tuning method for ant colony optimization for scheduling jobs in grid environment. *International Journal of Intelligent Systems and Applications*, 11(3):11.

Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., and Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. *Ieee Access*, 10:10031–10061.

Smith, T. C. and Frank, E. (2016). Introducing machine learning concepts with weka. *Statistical genomics: Methods and protocols*, pages 353–378.

Sukstrienwong, A. (2017). A Genetic-algorithm Approach for Balancing Learning Styles and Academic Attributes in Heterogeneous Grouping of Students. *International Journal of Emerging Technologies in Learning (iJET)*, 12(03):4–25.

Sung, H.-Y. and Hwang, G.-J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & education*, 63:43–51.

Taniguchi, Y., Gao, Y., Kojima, K., and Konomi, S. (2018). Evaluating Learning Style-Based Grouping Strategies in Real-World Collaborative Learning Environment. In Streitz, N. and Konomi, S., editors, *Distributed, Ambient and Pervasive Interactions: Technologies and Contexts*, Lecture Notes in Computer Science, pages 227–239. Springer International Publishing.

Teodorović, D., Nikolić, M., Šelmić, M., and Jovanović, I. (2022). Bee colony optimization with applications in transportation engineering. In *Advances in Swarm Intelligence: Variations and Adaptations for Optimization Problems*, pages 135–152. Springer.

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, 85(6):317–325.

Van den Bergh, F. and Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information sciences*, 176(8):937–971.

Velmurugan, T. and Anuradha, C. (2016). Performance evaluation of feature selection algorithms in educational data mining. *Performance Evaluation*, 5(02).

Veloz, A., Weinstein, A., Pszczolkowski, S., Hernández-García, L., Olivares, R., Muñoz, R., and Taramasco, C. (2019). Ant colony clustering for roi identification in functional magnetic resonance imaging. *Computational Intelligence and Neuroscience*, 2019.

Wang, S.-L. and Hwang, G.-J. (2012). The role of collective efficacy, cognitive quality, and task cohesion in computer-supported collaborative learning (cscl). *Computers & Education*, 58(2):679–687.

Xu, H. and Kim, M. (2024). Combination prediction method of students' performance based on ant colony algorithm. *Plos one*, 19(3):e0300010.

Yadav, R. S. (2020). Application of hybrid clustering methods for student performance evaluation. *International Journal of Information Technology*, 12(3):749–756.

Yang, Z., Liu, L., Li, N., and Li, H. (2024). A self-decision ant colony clustering algorithm for electricity theft detection. *Engineering Applications of Artificial Intelligence*, 133:108442.

Yasear, S. A. and Ku-Mahamud, K. R. (2021). Fine-tuning the ant colony system algorithm through harris's hawk optimizer for travelling salesman problem. *Int. J. Intell. Eng. Syst*, 14:136–145.

Zang, X., Jiang, L., Ding, B., and Fang, X. (2021). A hybrid ant colony system algorithm for solving the ring star problem. *Applied Intelligence*, 51(6):3789–3800.

Zheng, Y., Li, C., Liu, S., and Lu, W. (2018). An improved genetic approach for composing optimal collaborative learning groups. *Knowledge-Based Systems*, 139:214–225.

Zheng, Y., Liu, Y., Lu, W., and Li, C. (2016). A hybrid PSO-GA method for composing heterogeneous groups in collaborative learning. In *2016 11th International Conference on Computer Science & Education (ICCSE)*, pages 160–164, Nagoya, Japan. IEEE.

University of Sfax
National Engineering School of Sfax
Doctoral school Sciences & Technologies

Universidad de Las Palmas de Gran Canaria
Programa de Doctorado en Empresa, Internet y Tecnologías de las Comunicaciones

Abstract

Creating effective learner groups is a crucial factor in ensuring the success and fluidity of collaborative learning environments. One of the major challenges in clustering is the assignment of elements that do not naturally belong to any predefined cluster.
This study introduces a Hybrid Bio-inspired Ant Colony Clustering Approach for Constructing Collaborative Learning Teams, aimed at forming balanced and synergistic student groups. The proposed hybridization between the Ant Colony Clustering Algorithm and K-means effectively resolves the issue of ungrouped students by assigning them to appropriate existing clusters or creating new ones when necessary.
Experimental results demonstrate that dataset size significantly impacts clustering performance, particularly for Ant Colony-based algorithms that are sensitive to data distribution. The deterministic characteristics of K-means enhance the process by refining cluster boundaries and improving overall robustness.
Moreover, the Ant Colony-based algorithm, inspired by natural ant behaviors in nest organization and corpse clustering, is highly dependent on its parameter settings. In this context, the influence of the similarity parameter α was thoroughly examined. Findings reveal that the performance of Ant Colony algorithms is strongly influenced by α, as well as by the size of the datasets used.
Overall, the proposed hybrid model exhibits enhanced adaptability, scalability, and stability, consistently outperforming traditional clustering algorithms across varying dataset sizes.

Resumen

La creación de grupos de estudiantes eficaces es un factor crucial para garantizar el éxito y la fluidez de los entornos de aprendizaje colaborativo. Uno de los principales desafíos de la agrupación es la asignación de elementos que no pertenecen naturalmente a ningún grupo predefinido.
Este estudio presenta un enfoque híbrido de agrupación en colonias de hormigas, bioinspirado, para la construcción de equipos de aprendizaje colaborativo, con el objetivo de formar grupos de estudiantes equilibrados y sinérgicos. La hibridación propuesta entre el algoritmo de agrupación en colonias de hormigas y K-means resuelve eficazmente el problema de los estudiantes desagrupados, asignándolos a grupos existentes apropiados o creando nuevos cuando sea necesario.
Los resultados experimentales demuestran que el tamaño del conjunto de datos afecta significativamente el rendimiento de la agrupación, especialmente para los algoritmos basados en colonias de hormigas sensibles a la distribución de datos. Las características deterministas de K-means mejoran el proceso al refinar los límites de los grupos y mejorar la robustez general.
Además, el algoritmo basado en colonias de hormigas, inspirado en el comportamiento natural de las hormigas en la organización de nidos y la agrupación de cadáveres, depende en gran medida de la configuración de sus parámetros. En este contexto, se examinó exhaustivamente la influencia del parámetro de similitud α. Los resultados revelan que el rendimiento de los algoritmos Ant Colony está fuertemente influenciado por α, así como por el tamaño de los conjuntos de datos utilizados.
En general, el modelo híbrido propuesto presenta mayor adaptabilidad, escalabilidad y estabilidad, superando consistentemente a los algoritmos de agrupamiento tradicionales en conjuntos de datos de distintos tamaños.