IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 September 2023, accepted 7 November 2023, date of publication 14 November 2023,
date of current version 27 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332818

RESEARCH ARTICLE

Educational Data Mining Clustering Approach: Case Study of Undergraduate Student Thesis Topic

ANDRE[1,2], NANIK SUCIATI[1], (Member, IEEE), HADZIQ FABROYIR[1], (Member, IEEE), AND ERIC PARDEDE[3], (Senior Member, IEEE)

[1]Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2]Department of Informatics, Faculty of Engineering, Universitas Surabaya, Surabaya 60293, Indonesia
[3]Department of Computer Science and Information Technology, La Trobe University, Bundoora, VIC 3086, Australia

Corresponding author: Nanik Suciati (nanik@if.its.ac.id)

This work was supported by the Ministry of Education, Culture, Research, and Technology, Indonesia, under grant of the Doctoral Dissertation Research year 2022.

ABSTRACT This study aims to investigate the potential of educational data mining (EDM) in addressing the issue of delayed completion in undergraduate student thesis courses. Delayed completion of these courses is a common issue that affects both students and higher education institutions. This study employed clustering analysis to create clusters of thesis topics. The research model was constructed using expert labeling to assign each thesis title to a computer science ontology standard. Cross-referencing was employed to associate supporting courses with each thesis title, resulting in a labeled dataset with three supporting courses for each thesis title. This study analyzed five different clustering algorithms, including K-Means, DBScan, BIRCH, Gaussian Mixture, and Mean Shift, to identify the best approach for analyzing undergraduate thesis data. The results demonstrated that k-means clustering is the most efficient method, generating five distinct clusters with unique characteristics. Furthermore, this study investigated the correlation between educational data, specifically GPA, and the average grades of courses that support a thesis title and the duration of thesis completion. Our investigation revealed a moderate correlation between GPA, thesis-supporting course average grades, and the time to complete the thesis, with higher academic performance being associated with shorter completion times. These moderate results indicate the need for further studies to explore additional factors beyond GPA and the average grades of thesis-supporting courses that contribute to delays in thesis completion. This study contributes to the understanding and evaluation of educational outcomes within study programs, as defined in the curriculum, particularly concerning the design and implementation of thesis topics. Additionally, the clustering results serve as a foundation for future research and offer valuable insights into the potential of EDM techniques to assist in selecting appropriate thesis topics, thereby reducing the risk of delayed completion.

INDEX TERMS Computing classification system, undergraduate thesis, clustering analysis, k-means, ontology.

I. INTRODUCTION

Educational Data Mining (EDM) involves data mining, machine learning, and statistical methodologies to extract valuable insights from educational datasets [1]. These datasets are often obtained from Learning Management Systems (LMS) and include detailed information such as assignment submission dates, LMS access logs, and social interaction within the platform. Additionally, less detailed data, such as student transcript historical data, which contain information on courses attended and grades received, can also be used in the EDM. By analyzing these datasets, the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhigao Zheng.

EDM can identify trends, patterns, and relevant information that may not be immediately apparent, allowing for a deeper understanding of educational processes and outcomes.

In the field of education, EDM has become increasingly important for both learners and educators. Recent research has focused on applying advanced EDM techniques to analyze large datasets and extract meaningful patterns. EDM can be used to predict students' learning behavior [2], discover hidden information through clustering approaches [3], [4], [5], analyze the impact of learning methods, and advance scientific understanding. This technique can be applied to anomaly detection, association rule problems, clustering, classification, regression, and summary problems.

The clustering technique is an unsupervised learning method for obtaining information from a large dataset and studying the relationships and patterns between data. In clustering, data are grouped based on the proximity or similarity of their attributes. Studies have found that clustering techniques in education deliver benefits for the learner by delivering better recommendations, such as adjusting learning styles, material selection, educator selection, and other benefits to improve stakeholder performance [4], [5], [6]. The clustering technique is also applicable to EDM. Romero and Ventura's taxonomy [7] provides insights for numerous studies in this area. For example, the problems of determining significant contributors that affect learner performance [8], [9], the development of student learning profiles based on learner behavior data [6], [10], [11], and the prediction of miscellaneous academic outcomes (student dropout, learner performance, and learner behavior) [12], [13], [14], [15].

Several methods and points of view identify and investigate issues pertaining to the educational domain [16]. Universities in Indonesia officially refer to the government regulation of undergraduate thesis duration, which is a six-month timeframe specified in the course syllabus. This type of course has characteristics that differ from those of regular courses. The learning process involves mentoring supervisors/advisors and students, working on real-world topics, and requiring students' cognitive abilities. Students must tackle challenge critically, creatively, and independently for their undergraduate thesis to succeed. Nevertheless, a delay in completing an undergraduate thesis is one of the issues where students, on average, finish the thesis in more than six months or two semesters.

In Educational Data Mining (EDM), researchers often use clustering methods for various tasks such as classifying courses, predicting student behavior, and creating course recommendation systems. However, there remains a gap in the application of EDM techniques for categorizing undergraduate thesis topics. The selection of a thesis topic is of great importance to students, as it can affect their academic performance and time management, especially in their final year of study. As a result, we propose to investigate the most effective clustering results using historical data from undergraduate theses. What sets our research apart is the data preparation techniques that we employed to generate high-quality clusters of undergraduate student theses. This contributes to a better understanding and assessment of the educational outcomes defined in study programs related to the design and execution of thesis topics. Furthermore, our study explored the correlation between students' GPAs and the average grades in thesis-supporting courses concerning the time required to complete their theses. The results of this research can be beneficial for further studies related to topic or thesis title recommendation systems that consider students' academic transcripts. Therefore, the outcomes of this further research can assist students in selecting the right thesis topic or title, thereby minimizing delays in completing their theses.

The data labeling process focuses on expert judgment using the ACM Computing Classification System (CCS) as the standard for domain knowledge ontology in computer science. This technique uses expert judgment to select at least three courses that contribute to the undergraduate thesis title. However, to the best of our knowledge, a data preprocessing method based on domain knowledge derived from ontology has never been proposed. Therefore, the urgency of this research lies in the fact that the correct topic of the undergraduate thesis will significantly determine student success. The primary objective of this research is to examine how EDM can be applied to the analysis of previous undergraduate student thesis titles to uncover patterns and structures, leading to the identification of appropriate and accurate thesis topics through cluster analysis.

II. LITERATURE REVIEW

This section describes the literature on the benefits and applications of EDM, clustering-based approaches, clustering algorithms, and evaluation metrics.

A. EDUCATIONAL DATA MINING

EDM is a cutting-edge paradigm for establishing ways to evaluate atypical sources of evidence in educational environments [17]. Furthermore, it is essential to employ these methods to comprehend students and their learning settings. Classification, association, clustering, regression, forecasting, sequencing, and descriptive data-mining techniques are still used and exploited in EDM [18]. In addition, the EDM uses statistics and machine learning to enhance its effectiveness. Nonetheless, the fundamental goal of EDM is to discover knowledge from a collection of educational data that will benefit its stakeholders, primarily educators and learners [7].

The EDM is widely used to assess and understand students' motivations, attitudes, and behavior. For example, Rohlíkov [16] conducted research assessing student attitudes toward quizzes on the Moodle LMS. The dataset comprises 610 student activities from five Moodle quizzes. This study identified the reliability of the process mining method used to detect student attitudes during quizzes. EDM can predict motivational deficits in the classroom by focusing on the relationship between learning attitudes and student

performance [19]. The questionnaire was administered to 180 students from 48 different courses at six different universities. It generates a motivation index that divides students into three categories: autonomous (those who learn through their activities in the LMS), controlled (students who update their data and information in the LMS), and e-learning driven (those who learn through their activities in the LMS such as forums). The findings revealed a direct relationship among student performance (student results), autonomous groups, and e-learning motivation.

Another advantage of the EDM is that it allows the creation of a student profile model. Adaptive learning refers to a learning method that adjusts to the characteristics of each student, is more efficient, and has a more significant impact than conventional learning [20]. Kausar et al. [11] tested multiple clustering approaches on a dataset of 600 students and used a clustering technique to group students based on their historical data and learning behavior. The student profiles were developed using a clustering process and were used as the basis for the construction of personalized e-learning. On the other hand, Miranda et al. [21] revealed that EDM can be used to predict student dropouts. This study included 3,362 students with 51 features consisting of student academic records, family data, school characteristics, and admission process. It implemented data-driven adjustments to the educational environment to better map and classify students. This study revealed the profile of at-risk students, which can be used for early intervention. In conjunction with the students' perspective, the construction of a teacher profile has also been investigated in previous research. For example, Tondeur et al. [22] studied teacher profiling using questionnaire and association rule methodologies to identify the characteristics of effective teachers. This study found that trained teachers with more positive views placed greater emphasis on collaboration, whereas those with negative attitudes placed greater emphasis on feedback.

B. CLUSTERING APPROACH IN EDM

Cluster analysis is a data-mining technique used to group entities that share common traits compared to other entities belonging to other groups. This application is widely known for pattern recognition, multimedia retrieval, machine learning, and statistics, and is applicable to many subjects. Although many clustering algorithms exist, one of the most popular and frequently used algorithms is the k-means algorithm. This clustering technique groups objects into clusters using the nearest mean. The k-means algorithm iteratively divides the dataset into k clusters so that each node will have a minimum sum of the squared distance to its respective centroid.

A common application of the clustering technique with k-means is a segmenting system prevalent in the education domain. Using e-learning data, Rawat and Dwivedi demonstrated how a clustering technique combined with the k-means algorithm can categorize students based on their features and behaviors [5]. Moodle was used to collect student usage data on assignments and quizzes. The students' interactions can be retrieved from several sources such as forums, chat, and messaging while doing quizzes and assignments in Moodle. These data were generated as log files on the Moodle server, which were later extracted and preprocessed. Their model generated three clusters that depicted student profiles: non-active, average, and active. The number of clusters was then verified using elbow and silhouette evaluation, which is a heuristic metric used to determine the number of optimum clusters. They then created a course recommendation system on the Moodle platform, which produced results based on a cluster of student profiles. Finally, they implemented statistical metrics to evaluate the results, such as root mean squared error (RMSE), precision, recall, and F1. The conclusion stated that future research should explore extracting implicit ratings from Moodle server log files to enrich the user-item rating matrix. This solution will help overcome the challenge of sparse data from users who do not provide detailed ratings owing to a lack of motivation or incentives. Domain knowledge can also be extracted and integrated into the recommendation process to further enhance the learner profile and improve the quality of the recommendations.

Additionally, one of the main areas for improvement in building a course recommendation system using a k-means clustering algorithm with data extracted from the LMS is the need for more personalization in the recommendations. This is because the algorithm relies solely on clustering patterns in the data and does not consider the unique preferences and needs of individual students. Furthermore, the quality of the recommendations is highly dependent on the quality of the data, which can be affected by various factors, such as incomplete or inaccurate student profiles, biased or outdated data, and limited data available for specific user groups.

Aher and Lobo combined k-means clustering with an association rule algorithm to provide optimal course selection recommendations [23]. The dataset used for the analysis consisted of course enrolment data from 100 distance learning students, which were processed using the k-means clustering technique to form n-clusters. The experiments used three different clustering methods: Simple K-means clustering, Farthest First clustering, and the Expectation Maximization clustering algorithm. The association rule algorithm was then employed to determine the relationship between courses within the same cluster. The algorithm demonstrated that courses are more likely to be taken together and can be modeled through association rules. Furthermore, the results indicated that the Simple K-means clustering and Apriori association rule algorithm combination did not require the data preparation stage and produced more association rules, which increased the strength of the association rule. Future work includes exploring other combinations of data-mining techniques for course recommendations in distance learning, integrating the system with existing e-learning platforms, and potentially applying the system to MOOCs. Although

combining k-means and association rules yields better candidate courses than using association rules alone, one of the key disadvantages of the association rule algorithm is the lack of context in the correlations found. Association rule algorithms only focus on finding correlations between items and must consider the context in which these items occur, potentially leading to incorrect conclusions and rules that could be more meaningful. To address this issue, incorporating ontologies of knowledge into the analysis can provide a more contextual approach and better account for the context in which correlations occur, leading to more accurate conclusions and meaningful rules. In this case, the researcher could have considered the curriculum structure, since some courses may have prerequisites before the student can enroll.

Moubayed et al. [4] emphasized the importance of analyzing student engagement levels on e-learning platforms through clustering. The research was based on 486 undergraduate science students, and their activities were recorded in a student log. The researchers established an engagement meter to quantify student involvement by measuring their interaction and effort. Metrics related to interaction described how often the students engaged with the course content on the learning platform. In contrast, metrics related to effort described the level of exertion that the student put in to finish the course assignments. The parameters were event date, type, location, start time, end time, and student ID. The optimal number of clusters was determined through evaluation, ranging from two to five, based on prior literacy studies on engagement-level classification. The study was then analyzed using silhouette methods, and it was found that the number of clusters representing low and high engagement levels was considered the best result. Future studies should test the model on a different course/semester to investigate its generalizability, collect and evaluate the total time spent and average time per session to better gauge students' engagement, examine the impact of engagement metrics on student performance, and explore qualitative-based data analysis to modify course content based on student preferences. Another work currently under preparation explores the identification of weak students based on their course performance. However, it is essential to note that before students enroll in a course, researchers should thoroughly investigate the potential disadvantages of their engagement levels. Choosing an incorrect course can lead to demotivation and negative learning experiences. Clustering students' prior studies can help determine the best courses for each student, considering their engagement levels and learning styles. This personalization can lead to more personalized and compelling learning experiences, and increase the likelihood of success. Therefore, students must make informed decisions regarding their course selection to ensure optimal engagement and success.

Another study on learning behavior examined student migration patterns regarding the conformity of courses taken with curriculum guidelines [24]. The clustering technique is based on a limited set of educational and academic records such as grades, courses, IP, and timestamps. These preferences require factors that comprehensively influence student migration patterns. Additionally, using k-means algorithms to cluster similar objects in the education domain may not be the most appropriate method for effectively capturing the complex relationships between student migration patterns and curriculum conformity. The proposed P-CEA method for analyzing dynamic educational data can be improved by integrating demographic data, conducting social network analysis, developing a predictive model for identifying at-risk students, exploring cross-disciplinary applications, and refining the method by adjusting weightings or incorporating additional clustering algorithms. These future studies could enhance the understanding of factors contributing to student success or failure and provide early warnings and counseling to prevent students from dropping out.

Additionally, the study could benefit from incorporating ontologies into computer science to address these limitations. Ontologies provide a structured and standardized representation of knowledge that can be used to effectively capture the complex relationships between different entities. By incorporating ontologies, this study could better capture the factors influencing student migration patterns, such as student background, academic interests, and socio-economic factors. Additionally, using ontologies would provide a more comprehensive representation of the data, making it easier to analyze and interpret the results. This technique provides a more in-depth understanding of the relationship between student migration patterns and curriculum conformity.

Different clustering analysis studies on EDM have been conducted extensively in recent years. However, more work needs to be done on undergraduate thesis datasets. An undergraduate thesis is a comprehensive research project completed by students in their final year of study. It typically involves an in-depth investigation of a research question or topic of the student's choice, and the analysis and interpretation of data. Cluster analysis is a powerful data-mining technique that can be used to identify patterns and similarities in large datasets. Applying cluster analysis to undergraduate thesis projects can reveal insights that may only be apparent through traditional analysis methods such as identifying common themes or trends across multiple thesis projects. This information can inform future undergraduate students' curriculum design and research topics, and identify potential areas for further research. Additionally, cluster analysis can help students to better understand the broader context of their research and how it relates to similar research in their field. Ultimately, using cluster analysis to analyze undergraduate thesis projects can help identify valuable insights and inform future research directions.

C. CLUSTERING ALGORITHM

Clustering is a machine learning technique that groups data items according to their similarity or distance. The purpose of clustering algorithms is to split the data into groups or clusters, where each group contains comparable data points and is unique. Clustering algorithms facilitate exploration and

comprehension of detailed information by grouping similar data points. They can compress large datasets, detect anomalies, assist with recommendation systems, segment images, and identify market segments. Clustering techniques provide non-obvious insights into the patterns, trends, and correlations within the data. They can be utilized in numerous fields including data mining, machine learning, image processing, and marketing.

Numerous clustering methods include centroid-based, hierarchical, density-based, and distribution-based methods [25]. Centroid-based clustering is a clustering algorithm that combines similar data points based on their proximity to the centroid, which serves as the representative point of the cluster. In this clustering process, the algorithm allocates each data point to the nearest centroid after randomly selecting K centroids (where K is the desired number of clusters). The program then calculates the new centroids as the mean of all data points in each cluster and repeats the process of assignment and recalculation until convergence is reached.

K-means clustering and mean-shift algorithm are two of the most commonly used methods for clustering based on centroids. K-means minimizes the sum of the squared distances between each data point and its assigned centroid. The algorithm updates the centroids and iteratively reassigns the data points until convergence is achieved. K-means is computationally efficient, making it suitable for big datasets, and has been implemented in several applications, such as image segmentation, document clustering, and customer segmentation. The mean-shift algorithm is another centroid-based clustering algorithm that iteratively shifts each data point toward the most significant density of data points until it reaches a convergence point, which acts as the centroid of the cluster. The algorithm estimates the thickness of the data points using a kernel density function and the shifting procedure moves each data point in the direction of the steepest climb of the density function. Mean-shift is good at identifying clusters of different forms and sizes, making it appropriate for clustering applications, such as image segmentation and object tracking.

Both the K-means and Mean-shift algorithms have advantages and disadvantages. K-means clustering is sensitive to the initial selection of centroids, resulting in various clusters. However, mean-shift is computationally more expensive than K-means, rendering it unsuitable for large datasets. Moreover, mean shift may yield an arbitrary number of clusters, and determining the appropriate number of clusters can be challenging. Despite these drawbacks, centroid-based clustering is widespread, owing to its relative efficiency and effectiveness in discovering clusters in high-dimensional data.

Hierarchical clustering is a technique that groups data points with similar characteristics based on their proximity. With this type of clustering, the algorithm generates a hierarchy of clusters, beginning with individual data points as the initial clusters and merging them iteratively until all the data points belong to the same cluster. The two primary types of hierarchical clustering are agglomerative and divisive clustering. The agglomerative begins with each data point as its cluster and continues by combining the most comparable clusters until all the data points belong to a single cluster. In contrast, divisive clustering begins with all the data points in a single cluster and recursively divides them into smaller groups. The algorithm determines the similarity between clusters or data points using a distance metric in all the hierarchical clustering methods. Several metrics, such as Euclidean distance, cosine distance, and correlation distance, can be used to calculate the distance between two points.

Balanced Iterative Reduction and Clustering (BIRCH) is a common hierarchical clustering technique developed to cluster massive datasets effectively. BIRCH uses a tree-based data structure to represent data points and clusters, thereby allowing it to progressively create and update the clustering model as new data points are introduced. Additionally, the technique employs a clustering mechanism that compresses the data points, reduces memory requirements, and enables BIRCH to handle large datasets efficiently. The branching factor, threshold number, and number of clusters are three critical factors that can be modified to maximize BIRCH's clustering performance of the BIRCH. The branching factor sets the maximum number of child nodes associated with each internal node in the tree. The threshold value defines the maximum number of data points that an internal node can carry before splitting it into two child nodes. Finally, the number of clusters determines the desired number for the output.

Density-based clustering is a technique that clusters densely packed data points while isolating less dense regions. This clustering technique is excellent for detecting clusters of arbitrary shapes, and can handle noise and outliers. DBSCAN is the most prevalent density-based clustering algorithm (density-based spatial clustering of noisy applications). DBSCAN operates by identifying dense regions of data points and allocating them to the same cluster. The algorithm requires two parameters: the minimum number of data points needed to build a dense region (called minPts), and a distance measure that determines the radius surrounding each data point within which other data points are considered neighbors.

DBSCAN begins by randomly selecting an unvisited data point and determining if it has at least minPts neighbors within a distance measure-defined radius. If a point has sufficient neighbors, it is placed in a new cluster. Otherwise, the point is labeled as noise or a boundary point, and the algorithm continues to the next unvisited point. Next, DBSCAN checks the neighbors of each newly added data point to a cluster, and adds them to the same cluster if they have sufficient neighbors within the radius. The procedure was repeated until all dense sections of data points were allocated to clusters and all noise or boundary points were found.

Data noise and outliers can be handled using DBSCAN and other density-based clustering techniques, which is an advantage. In addition, they may recognize clusters of arbitrary

shapes, which is challenging for existing clustering algorithms that assume that clusters are spherical or have a specific shape. However, a disadvantage of DBSCAN is that it requires careful parameter adjustment to produce optimal results, and the clustering outcome can be sensitive to the distance measure and minPts. In addition, this technique may perform poorly on datasets or clusters with drastically varying and fluctuating densities.

Distribution-based clustering algorithms assume that data points are created from a probability distribution and employ statistical methods to identify data groups. This clustering technique is excellent for detecting clusters that follow a specific distribution such as a Gaussian or Poisson distribution. The GMM is a popular approach for distribution-based clustering (GMM). The GMM implies that the data points are derived from a mixture of Gaussian distributions, with each cluster representing a different Gaussian component. Using an iterative technique such as expectation maximization, the process estimates the parameters of the Gaussian mixture model, such as the mean and covariance of each element (EM).

The GMM algorithm begins by randomly initializing the parameters of the GMM. Using Bayes' rule, the computer iteratively calculates the likelihood that each data point corresponds to each component of the Gaussian mixture model. Based on these probabilities, the algorithm modifies the parameters of the Gaussian mixture model to match the data better. This procedure is repeated until the algorithm reaches a solution. The GMM and other distribution-based clustering methods may need to perform better on datasets with irregularly sized or shaped clusters. In addition, they may require careful parameter tweaking and are sensitive to the number of components used in the mixture model. In general, distribution-based clustering is a robust technique that can handle various data types and applications, particularly when the data points follow a particular distribution.

D. CLUSTERING PERFORMANCE EVALUATION

Unlike supervised approaches, where the ground truth is used as an indicator of clustering performance evaluation, as an unsupervised approach, the clustering results obtained using k-means do not have a specific evaluation measure associated with them. In this case, because the number of clusters depends on the initial input, an approach for evaluating the performance of the model based on the number of clusters is required. These include the elbow method and metric evaluation, such as silhouette analysis, Calinski-Harabasz, and Davies-Bouldin score index.

The Elbow method indicates the optimum number of clusters based on the sum of the squared distances between the data points and cluster centroid. The results of the calculation are then plotted onto a diagram that resembles an ''elbow'' shape. A heuristic rule of thumb states that the optimal number of selected clusters is reached when the graph exhibits diminishing returns. The graph then moves approximately in a straight line parallel to the x axis. The K value that corresponds to this point is the optimal K value or ideal number of clusters.

The silhouette analysis metric can identify the quality and performance of cluster results. The silhouette coefficient determines the degree to which clusters are separated from one another. The formula for calculating the coefficients is shown in Equation (1).

$$s\left(o\right) = \frac{b\left(0\right) - a(0)}{\max\{a\left(0\right), b\left(0\right)\}} \tag{1}$$

where $a(0)$ denotes the average distance between point o and all other data points within its cluster. The $b\left(0\right)$ is the average distance between o and all clusters to which o does not belong and is expressed as the minimum average distance. A coefficient close to -1 indicates that the number of clusters is not optimal. A value close to zero indicated overlapping clusters. As a result, to construct the best cluster, it is often desired that the coefficient be significant and close to one.

Silhouette analysis evaluates the clustering quality of each data point by calculating the distance between the data point and other points in its cluster, as well as the distance between the data point and points in the neighboring cluster. Higher silhouette scores indicated superior cluster quality. Conversely, a high silhouette score suggests that a data point is well-matched to its cluster and poorly matched to nearby clusters, which suggests that clustering is effective.

The Calinski-Harabasz index is another cluster evaluation metric [26]. Clustering validation calculates the ratio of the sum distribution of data points within and between clusters. The Calinski-Harabasz calculation formula is given in (2).

$$s = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k - 1} \tag{2}$$

where s is the Calinski-Harabasz score resulting from the division of the dispersion ratio between clusters $tr(B_k)$ and the dispersion ratio within the cluster $tr(W_k)$. n_E refers to the number of data points and k is the number of clusters. Calinski-Harabasz evaluated the ratio between the cluster variation and within-cluster variance, reflecting how effectively the clusters were separated. A high Calinski-Harabasz score suggests that the clusters have unique patterns and a wide gap between their means.

The Davies-Bouldin metric is another clustering evaluation method that calculates the average similarity for each cluster compared with other similar clusters [27]. Davies-Bouldin calculated the average similarity between each cluster and its most similar cluster by calculating the distance between the cluster means and cluster sizes. In contrast to Calinski-Harabasz, a low Davies-Bouldin index suggests that clusters are well-separated and distinct, with limited overlap or similarity. The Davies-Boulding equation is given by (3).

$$S_{i,j} = \frac{P_i + P_j}{D_{i,j}} \tag{3}$$

P_i is the average distance from the data point to the centroid of the cluster i. The same applies to P_j. Meanwhile, $D_{i,j}$

represents the distance between the cluster centroids i and j. Therefore, the ratio between the average distance between two clusters i and j and the distance between the clusters is shown in the $S_{i,j}$ similarity value.

In conclusion, the silhouette metric examines the quality of the clustering of individual data points, the Calinski-Harabasz method analyzes the separation and distinctness of the clusters as a whole, and the Davies-Bouldin metric evaluates the similarity and overlap between the clusters. Depending on the unique objectives and characteristics of the clustered data, each indicator can provide valuable insights into the performance of clustering methods.

III. METHODOLOGY

This section is divided into distinct sections. The first part describes some underlying problems that contributed to the late completion of the student thesis. The second half presents the clustering architecture, and the last part explains clustering validation using different techniques.

A. UNDERGRADUATE THESIS PROBLEMS

It is envisaged that the final undergraduate thesis will be completed within six months, as specified in most course syllabi. Based on statistics, the students took more than six months to complete the thesis. For example, in a case study conducted at the Informatics Engineering Department of the University of Surabaya, 300 students completed their undergraduate theses during the graduation period of 2016-2021. The average amount of time required to finish the thesis was 8.5 months. It takes the shortest (3.13 months) and longest (24.36 months) times. Students typically spend two semesters on their undergraduate theses.

Completion time affects study duration as a contributing factor that determines the quality of the university, as quantified by the standard accreditation score. In addition, students who finish their studies on time have benefits in terms of study costs, scholarship considerations, and others.

The determinants of the delayed completion of undergraduate theses are motivation, cognitive abilities, and the supervisor's role [2], [28], [29]. Three factors contribute to low motivation.

a) Low autonomy: students do not like the topic of their final assignment, or they do not have room to make decisions [14], [30];

b) Low usefulness: students feel that the topics they are working on have no impact or are less useful.

c) General/academic procrastination: Students procrastinate, which has a small but cumulative impact on late completion [31], [32], [33].

The above factors contribute to low motivation in working on student theses, which would affect the delay in finishing their theses on schedule. Therefore, as a precautionary step to minimize the problems, students should be able to choose their appropriate thesis topic. In doing so, the student will benefit by having a range of suitable topics as their consideration to choose one as their preference. However, providing a range of relevant topics for students is challenging because the breadth and complexity of potential topics can be overwhelming and the scope may differ among universities. To address this problem, this study uses EDM techniques to analyze historical undergraduate thesis data and uncover hidden patterns. The goal was to determine suitable clusters of thesis topics that students could choose based on their interests and proficiency. This analysis utilizes the standard Computing Classification System (CCS) ontology, which categorizes fields in computer science into 13 primary knowledge domains with branches up to four levels of depth. By mapping each undergraduate thesis title to multiple knowledge domains, this study aimed to provide valuable information for students to make informed decisions about their thesis topic.

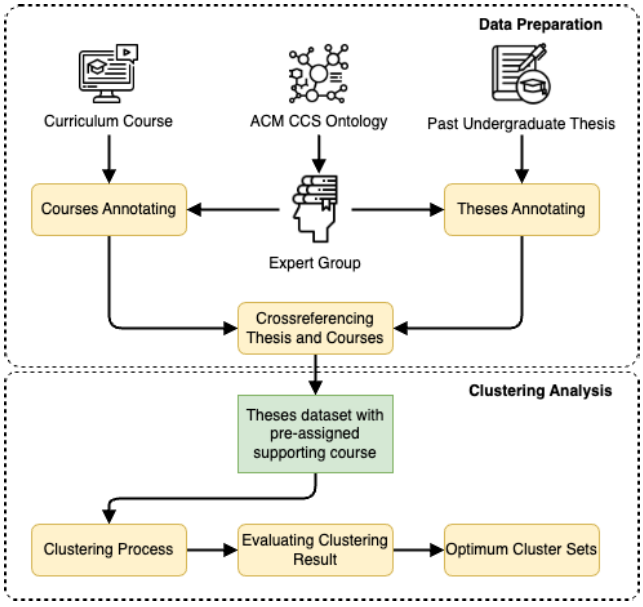

FIGURE 1. Student thesis topic clustering system architecture.

B. THE CLUSTERING SYSTEM ARCHITECTURE

We present a clustering system architecture of students' undergraduate theses with main novelties focused on the involvement of computer science ontology to determine the thesis base supporting knowledge curated by experts. Our methodology begins with data preparation, which involves expert decisions to annotate past undergraduate theses and support courses. The clustering process can then begin with a pre-annotated dataset. Finally, we investigated the proper clustering algorithm configuration within the clustering process step to produce optimum clusters. Fig 1 illustrates our system architecture.

1) DATA PREPARATION

The clustering system architecture begins with data preparation. Three data sources are involved: curriculum courses, CCS ontology, and past undergraduate theses. The curriculum course source highly depends on each institution's curriculum design; however, they should support the student's

TABLE 1. Computing methodologies CCS ontology snippet.

Top-level	2nd tier	3rd tier	4th tier
Computing methodologies			
	Symbolic and algebraic manipulation		
		Symbolic and algebraic algorithms	
			Combinatorial algorithms, Algebraic algorithms, Nonalgebraic algorithms
			. . .
		Computer algebra systems	
		. . .	
	Parallel computing methodologies		
		Parallel algorithms	
			MapReduce algorithms, Self-organization, Shared memory algorithms
		Parallel programming languages	
	Artificial intelligence		
		Natural language processing	
			Information extraction, Machine translation, Discourse, dialogue and pragmatics
		Knowledge representation and reasoning	
		. . .	
	. . .		

TABLE 2. Computing methodologies ACM CCS ontology snippet.

Course	ACM CCS	Coefficients
Modeling and Simulation	Modeling and Simulation	0.6
	Probability and statistics	0.3
	Mathematical analysis	0.1
Decision Support Systems	Operation Research	1
Big Data Analytics	Design and analysis of algorithms	0.5
	Visualization	0.3
	Machine learning	0.2
Artificial Intelligence for Game	Theory and algorithms for application domains	0.5
	Artificial intelligence	0.4
	Cross-computing tools and techniques	0.1

undergraduate thesis as a penultimate course within a degree. In our case, we employed 72 courses, consisting of 23 compulsory courses; the rest were elective courses. Our research only chose limited mandatory courses because we hypothesized that students had already mastered the introductory course upon reaching the final semester. In contrast, elective courses include more advanced subject matters and frequently cover multiple disciplines or competencies simultaneously.

Formal ontologies to establish ground knowledge of the forthcoming annotation use CCS, accessible online at https://dl.acm.org/ccs. CCS is used to classify research publications in the computer science field. We see that ontology is relevant to the needs of our system architecture and that the Computer Science curriculum in most universities worldwide has also adopted CCS in their curriculum design.

The final data source was students' undergraduate thesis records. We extracted 300 past undergraduate theses from the class of 2016–2021 in Informatics degree, Universitas Surabaya. Our dataset can be downloaded from the following repository: https://github.com/scancampy/student-thesis-dataset. In this study, all 300 titles contained different computers science knowledge areas, such as information management, computational science, intelligent systems, software engineering, graphics and visualization, and human–computer interaction. We aim to cluster undergraduate computer science topics and highlight each cluster's insights and characteristics.

Following data collection, the data preparation involved forming an expert group for analysis, as illustrated in Fig 1. The group comprised the laboratory head, supervisor, and curriculum design team. The experts undertake three activities, beginning with a study of the computer science domain hierarchy derived from CCS ontology. The relationship between the experts and the CCS data source is shown in Fig 1. The CCS hierarchy consists of knowledge area ontologies that extend to the fourth level. Each level contains knowledge areas of computer science, with the top-level hierarchies representing general knowledge areas and the deeper levels showing more specialized knowledge areas. Table 1 presents an example of the Computing Methodologies ontology hierarchy, one of the top-level hierarchies of CCS ontology, and provides more specialized knowledge areas at a deeper level.

We assign formal ontologies to each course to ensure alignment between the course and thesis requirements. Experts manually annotated each course by examining the terminology in the syllabus, lesson plans, and other relevant documents to arrive at appropriate decisions. We used the CCS ontology to label each course, as it contains a large amount of material for each general topic. Multiple ontologies may be related to each course and coefficients are used to determine their contributions. For example, in the Big Data Analytics course shown in table 2, the design and analysis of algorithm ontology has the highest coefficient, followed by Visualization and Machine Learning ontologies. By cross-referencing the thesis and course ontologies, we can identify courses that match the thesis requirements and determine which students *should* master. The coefficients indicate the extent to which a particular ontology contributes to the course content.

After annotating all 72 courses, the experts continued with the annotation of the thesis. To annotate thesis titles

TABLE 3. Annotation result data snippet.

Priority	1st tier	2nd tier	3rd tier	4th tier
High	**Operations research**			
	↳	Decision analysis		
		↳	Multi-criterion optimization and decision-making	
Medium	**Information systems applications**			
	↳	Decision support systems		
Low	**Software notations and tools**			
	↳	General programming languages		
		↳	Language features	
				Frameworks

with related CCS ontologies, experts evaluated each thesis document's title, abstract, and keywords to select relevant ontologies. For example, in table 3, the thesis titled "Development of Decision Supporting Systems Using the Weighted Product Methodologies for Credit Installment of Vehicle Sales" is annotated with three contributing ontologies based on their relevance to the title (high, medium, and low). Next, the expert selects the deepest branch of the ontology structure. As shown in table 3, the contributing knowledge areas are operational research, information system application, and software notation and tools. The three deepest and most relevant ontologies are multi-criteria optimization and decision-making, decision support systems and frameworks, the 3rd tier components rooted in operations research. Expert involvement strengthens the accuracy and reliability of the labeling outcomes. Utilizing this ontology, we examine the relevance of the thesis topic to the deepest ontology branch. The deeper the selected ontology, the more accurate the classification process.

2) ONTOLOGIES CROSSREFERENCING

We annotated the dataset using CCS ontologies to identify knowledge areas relevant to a particular thesis. We matched the ontologies of courses and thesis titles using a cross-referencing process, as shown in Fig 1. We developed expert annotation tools, a web-based system, to help experts conduct the annotation process for courses and theses [34].

Our expert annotation tools aid in the annotation process, making it semi-automatic. This means that the tools automatically select the three courses that contribute the most to a given thesis title, based on cross-referencing and coefficients. However, experts can still review and manually adjust the results, if needed. Fig 2 shows a screenshot of the annotation tool displaying the top three courses for a specific thesis title. Experts first annotated each thesis with relevant ontologies to determine the courses that contributed to the thesis title. We then used cross-referencing to identify all courses that share the same ontologies as the thesis. Each course contained different ontologies with varying coefficients, indicating their degree of contribution to the course content. The annotation tool sorted these courses in descending order of their

TABLE 4. Snippet of course encoding.

Encoding	Root Ontology	Course
1	Software and Its Engineering	Web Programming
2		Web Framework Programming
3		Full-Stack Programming
	...	
13	Networks	Computer Network
14		Distributed Programming
15		Advanced Computer Network
	...	
18	Human Centered Computing	Human Computer Interaction
19		Mixed Reality
20		Immersive Computing
	...	
53	Computing Methodologies	AI Fundamental
54		Machine Learning
55		Modeling and Simulation
	...	

coefficient values to determine the most significant contributors to the thesis. The tool then automatically selects the top three courses with the highest coefficients, which experts can review and manually adjust, if needed. However, because we involved more than one expert in annotating a single thesis, it may be common to appear that there are disagreements among the experts in selecting the courses. Our tools can highlight disputes by providing an easy interface and facilitating experts to vote [34]. In conclusion, our streamlined approach enabled experts to identify the most relevant courses for a given thesis title. We obtained a dataset of annotated thesis titles, each with three contributing courses that can be used for clustering analysis.

To facilitate the upcoming clustering analysis, we selected three courses that supported each thesis title as the dataset features. These features are critical for ensuring that the resulting clusters accurately reflect the knowledge areas covered in each thesis. We chose this because this thesis is a crucial component of a student's academic career, allowing them to showcase their knowledge and skills. To excel in their thesis, students must have a solid understanding of the various supporting theories and concepts.

By taking advanced and specialized courses, students can deepen their knowledge and skills beyond the introductory level [35]. For instance, in artificial intelligence, a student may take advanced courses on in-depth algorithms, such as genetic algorithms, or deep learning courses that focus on artificial neural networks. Additionally, most universities offer elective courses that students can take to explore their interests and expand their knowledge. For example, at Universitas Surabaya, where the case study was conducted, students typically take 3-5 electives. Based on these reasons,

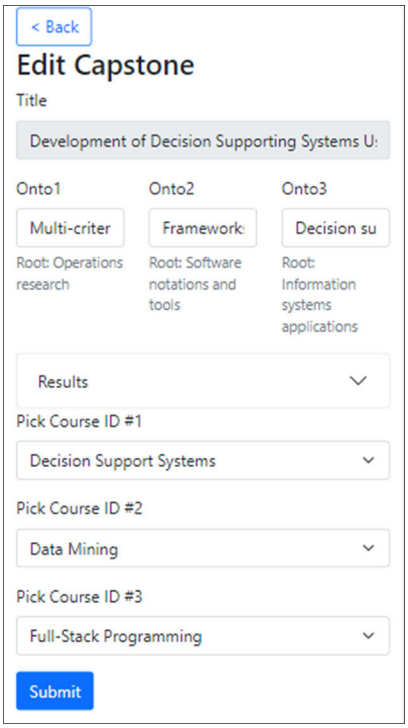

FIGURE 2. Expert annotation tools automatically determined three supporting courses.

we determined that each undergraduate thesis should have at least three supporting courses to enhance its fluency and comprehensiveness. This approach ensures that students have a solid foundation in relevant knowledge areas and can produce high-quality work.

3) IMPLEMENTATION OF CLUSTERING TECHNIQUES

Clustering analysis, especially an algorithm that uses distance metrics, requires a numerical representation of each data point, which is commonly achieved using encoding methods that assign specific values to each data point. However, a label encoding method is required for categorical data, such as our dataset, which refers to courses supporting individual thesis titles. Label encoding assigns a unique numerical value to each category in the dataset, allowing categorical data to be numerically represented for clustering analysis. Our study organizes courses based on knowledge areas using CCS ontology. Table 4 depicts the snippet of each course's encoding label to the numerical representation associated with the knowledge areas (root ontology). Our study demonstrates that the choice of encoding is not critical if it preserves the relative distance between data points and the clustering results should be similar.

In the following steps, we investigate the best clustering algorithm to deliver an optimal cluster set to extract features concealed from the view. We selected five clustering algorithms: k-means, mean shift, DBScan, BIRCH, and Gaussian mixture. K-means and Mean-shift are based on the centroid, whereas the rest are based on density, hierarchy, and distribution.

Different clustering methods can reveal a variety of traits. A clustering method known as density-based clustering groups data points that are concentrated in an area with high density. This clustering technique does not consider outliers and ensures that the cluster center point is located at the clustered data point. When performing distribution-based clustering, careful consideration is given to the probability that the algorithm includes a data point in the cluster. The further away a data point is from the cluster's epicenter, the lower the possibility that the algorithm will include it in the cluster. Calculating the squared distance from the predefined centroid indicates how each data point in the centroid-type cluster is created. Adjustments were made to determine the new centroid's location at the end of each iteration until the convergence criterion was met. Hierarchical clustering is a subtype designed solely for use with hierarchical datasets.

Among the popular centroid-based clustering algorithms, k-means and mean-shift have been established as solid algorithms that produce optimum cluster sets. In k-means clustering, we determined the number of K clusters for the algorithm process and delivery. However, K is not the best choice. We used the elbow technique, a heuristic approach, to determine the scoring index that defines the quality of the clusters' results. The degree of variance in each cluster number can be determined using the elbow approach, which involves calculating the square distance that separates each point from the center of the cluster. The steps of the k-means clustering process are as follows. First, the data encoding process converts the dataset value into a numeric representation. The clustering method can only read numeric data. Encoding values are organized into groups according to the extent of their underlying scientific basis. For example, data science and artificial intelligence courses use encoding values in the range of 1–50. Software engineering and enterprise system courses use an encoding value of 50–100. This procedure was applied to all fields. Second, the number of clusters was determined. This number is expressed as the optimal number, as proven using the elbow method in the Results and Discussion section. Finally, the clustering process was conducted. The algorithm runs iteratively until convergence is accomplished and all data points are appointed to the nearest optimum cluster center.

The Mean-Shift technique, which is another centroid-style clustering algorithm, is an alternative to the k-means technique. This algorithm uses unsupervised learning without the need for any parameters. This algorithm first computes the mean of the dataset and then shifts each data point to the area of the cluster mean that is closest to the center of that mean. Shifting this value does not change the original value but only keeps the label. In most cases, the mean shift performs well for image datasets [36].

As in the case of the density-based DBScan algorithm, the two most important factors are eps and the minimum data point (minPts). The eps parameter is used to configure the maximum distance that can exist between two data points before those points are no longer considered part of the

TABLE 5. Clustering technique configuration.

Technique	Based	Configuration
K-Means	Centroid	`Number of cluster = 5, kmeans++`
DBScan	Density	`Eps = 7, minPts = 30`
BIRCH	Hierarchical	`Branching factor = 50, threshold =7, cluster =5`
Mean-Shift	Centroid	`N/A`
Gaussian Mixture	Distribution	`n_component = 9`

neighborhood. In contrast, the minPts parameter specifies the least number of data points that must be present in a cluster for it to be considered valid. Some researchers have proposed an automatic method for determining these parameters [37], [38], [39].

Furthermore, the BIRCH clustering method is effective for large amounts of data. This method condenses the dataset into a succinct summary while maintaining as much of the information as possible. To reduce the time required to complete the operation, a clustering procedure was applied to the compact dataset version. The branching factor, threshold, and number of clusters are all examples of BIRCH parameters. The branching factor is the maximum number of CF subclusters that can be found on each individual node. The maximum number of data points that can be contained within a subcluster of the CF Tree's leaf node is referred to as the threshold. Cluster n is the anticipated total number of target clusters that will exist once the BIRCH algorithm has been run to completion. The BIRCH parameter can be determined automatically [40].

The Gaussian Mixture Technique is a clustering algorithm that employs a distribution-based approach. This algorithm performs a clustering process similar to that of k-means clustering. The Gaussian Mixture differs from k-means in that it considers the distribution and covariance of the data distribution. This allows the visual shape of the algorithm outputs to change, as opposed to k-means, which often produces a circular output. Both hard and soft clustering can be accomplished with the help of this approach. In contrast to hard clustering, soft clustering assigns a probability to each data point based on whether it belongs to a cluster. The parameter known as the n component is used by the Gaussian Mixture algorithm, which specifies the number of clusters that was produced using this method. To determine the number of clusters, we used the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), which evaluate the complexity of the dataset.

IV. RESULT AND DISCUSSIONS

This section is divided into two sections. The first part explains the clustering results obtained using popular clustering techniques. The second part presents the results of clustering based on ontology content and the correlation between clustering results and GPA, as well as the supporting course grades.

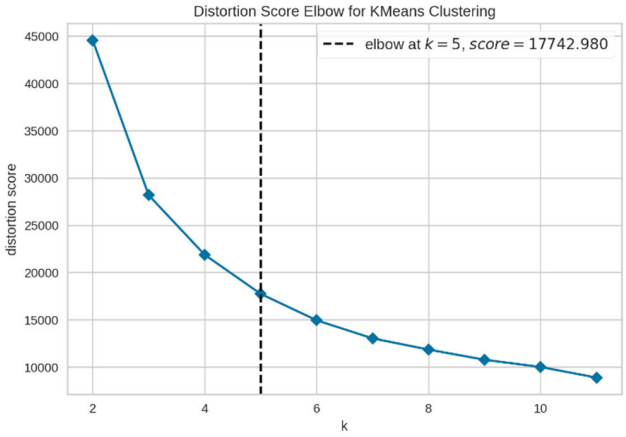

FIGURE 3. Elbow metric graph present the number of optimum cluster.

A. EXPERIMENTING WITH CLUSTERING TECHNIQUES

The results of the data preprocessing are 300 thesis titles that are ready for further processing through the clustering algorithm. The clustering process used Python, the Scikit library, and the Google Collab Notebook platform. The dataset synthesizes 300 thesis titles that have undergone data pre-processing. We performed an optimal configuration for each clustering technique to conduct the clustering process. Before the clustering process, we ensured that we used the most effective configuration for each type of clustering. The configuration of each clustering method is presented in Table 4.

We conducted the first experiment using k-means clustering with a dataset that had been annotated by three different supporting courses. As a result, our k-means clustering algorithm identified five clusters. The confirmed value can be demonstrated with complete confidence using the elbow measurement method, as depicted in Fig 3. This is due to the fact that the cut-off points for any number of clusters above five is regarded to have converged and increasing the number of clusters does not significantly alter the results. In addition, another centroid-based clustering algorithm called mean shift does not require configuration because it is a non-parametric unsupervised learning algorithm and does not account for any cluster or feature shapes.

For the DBScan, we used two parameters: MinPts and eps. First, we conducted trial and error by experimenting with various MinPts and eps values to produce the best possible clustering result. MinPts indicates the minimum number of data points required to determine a cluster. After determining the MinPts, a range of values for eps was tested to find the best clustering results. Using a MinPts value of 30 and ep value of 7 resulted in the best clustering performance. We also applied this manual testing method to the BIRCH algorithm and found that a branching factor of 50, threshold of 7, and a total of five clusters produced the best results.

TABLE 6. Result of clustering technique performance.

Technique	Num. of Cluster	Execution Time (seconds)	Silhouette score	Calinski-Harabasz score	Davies Bouldin score
K-Means	5	0.03181781769	**0.4206**	**190.1684**	0.858
BIRCH	5	**0.01628289223**	0.3480041781	133.9241321	**0.6055691236**
Gaussian Mix	9	0.06254787445	0.405	140.626	0.901
DBScan	2	0.1276900768	0.103	11.198	3.738
Mean-Shift	3	1.660001612	0.395	98.309	0.946

TABLE 7. Statistical summary of cluster based on ACM CCS root ontology.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Mean
Software Engineering	**48.00%**	**37.28%**	**46.67%**	31.01%	0.00%	**35.32%**
Networks	0.44%	3.23%	0.00%	0.00%	0.00%	1.10%
Human Computer Interaction	8.44%	2.51%	20.00%	1.55%	0.00%	4.64%
Theory of Computation	9.78%	32.26%	0.00%	0.78%	2.38%	12.80%
Mathematics of Computing	0.00%	0.72%	5.00%	1.55%	8.33%	1.77%
Information System	18.22%	24.01%	18.33%	**37.60%**	38.10%	27.37%
Computer System & Organization	0.44%	0.00%	0.00%	3.10%	7.14%	1.66%
Computing Methodologies	10.22%	0.00%	1.67%	18.22%	**41.67%**	11.70%
Applied Computing	3.56%	0.00%	8.33%	2.71%	2.38%	2.43%
Hardware	0.00%	0.00%	0.00%	3.10%	0.00%	0.88%
Security	0.89%	0.00%	0.00%	0.39%	0.00%	0.33%

Mixture Technique is a distribution-based method that requires the cluster number to be set up at the beginning. This number can then be determined by the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), which measures the complexity of the dataset and provides the ideal number of clusters, which in our case is five.

We used three commonly used quality metrics to measure the performance of different clustering methods on the undergraduate student thesis dataset: Silhouette score, Calinski-Harabaz index, and Davies-Bouldin index. The silhouette score measures the similarity of a point to its cluster compared to other clusters. The score ranges from -1 to 1, where a score of 1 indicates that the point is well matched to its cluster and poorly matched to neighboring clusters. A score of 0 indicates that the point is equally similar to neighboring clusters to its own cluster, and a negative score indicates that the point is more identical to neighboring clusters than its own. The Calinski-Harabaz index measures the ratio of between-cluster to within-cluster variance. Higher values indicate better-defined clusters, with larger separations between clusters, and more minor variances within each cluster. The Davies-Bouldin index measures the average similarity between each cluster and the most similar cluster. Lower values indicate better clustering with tighter and more separate clusters.

The results of the performance and evaluation metrics tests are presented in Table 6. Our analysis of various clustering methods found that K-means performed the best in the Silhouette score, Calinski-Harabasz, and Davies Bouldin metrics. K-means is a distance-based clustering algorithm that works well when the clusters have a spherical or circular shape, and the data points are well separated. However, BIRCH was the fastest algorithm with an execution time of 0.01628 s. This is because BIRCH is an algorithm that constructs a tree-based data structure to represent the data distribution and performs clustering on a condensed version rather than the entire dataset. Likewise, the Gaussian Mixture algorithm produced a relatively large number of clusters (nine) compared to the other algorithms. This is because Gaussian Mixture models are flexible and can model complex shapes of clusters. Hence, they fit the data better when the underlying distributions are complex or have multiple modes.

The clustering results indicate that both density-based algorithms (DBSCAN and Mean-Shift) produced relatively poor results compared to the other clustering algorithms. This is because our dataset has varying densities or irregularly shaped clusters, and is not concentrated. In datasets

with varying densities, choosing appropriate values for the algorithm's parameters, such as eps and minPts, may be difficult. Specifically, DBSCAN has a Silhouette score of 0.103, Calinski-Harabasz score of 11.198, and Davies Bouldin score of 3.738, indicating that the clusters are not well separated and overlap. The mean shift produces a silhouette score of 0.395, Calinski-Harabasz score of 98.309, and Davies Bouldin score of 0.946 with three clusters, but is the slowest clustering algorithm with an execution time of 1.6600 s. The Mean-Shift algorithm is a density-based clustering algorithm that is computationally expensive when dealing with large datasets. The slowness of the algorithm can be attributed to factors such as the bandwidth parameter, convergence criteria, and dataset size.

A short execution time is crucial when selecting an algorithm to ensure an optimal performance. As more data are regularly added, an efficient algorithm is imperative to ensure fast and accurate results. It is important to consider this factor in the algorithm-selection process. Following the preceding discussion, we conclude that in our case, computer science students' undergraduate thesis dataset would benefit from applying the k-means clustering technique.

However, this experiment showed that when applying clustering analysis, we must consider four aspects: dataset characteristics, understanding goals/research questions, using evaluation metrics, and scalability. Understanding the structure, size, and distribution of the data can help select a suitable clustering algorithm. Additionally, understanding the research question can help determine the most appropriate clustering algorithm. For example, density-based clustering algorithms such as DBScan may be more suitable for identifying outliers or anomalies. The clustering results should be evaluated using appropriate metrics, such as the Silhouette, Calinski-Harabasz, and Davies Bouldin scores. The performance of the clustering algorithm should be compared to that of other algorithms, and the results should be interpreted in the context of the research question. Some clustering algorithms may not be suitable for large datasets because of their computational complexity and memory usage. Therefore, scalability of the algorithm must be considered.

One advantage of clustering this dataset is its ability to examine the features of the thesis subjects chosen by students. In future research, we can use the results of this cluster as a component of a recommendation system. The expert already performs the annotation process using CCS ontologies as references and preserves the basics of determining the courses that support the thesis title. The recommendation system allows students who want to undergo thesis topics to choose courses from their transcript as a vital input recommendation system.

B. CLUSTER RESULTS

Table 7 displays the statistical distribution of computer science ontology in five clusters, categorized based on the ontology used to encode the previous dataset labels. The percentage of each ontology per cluster was calculated by

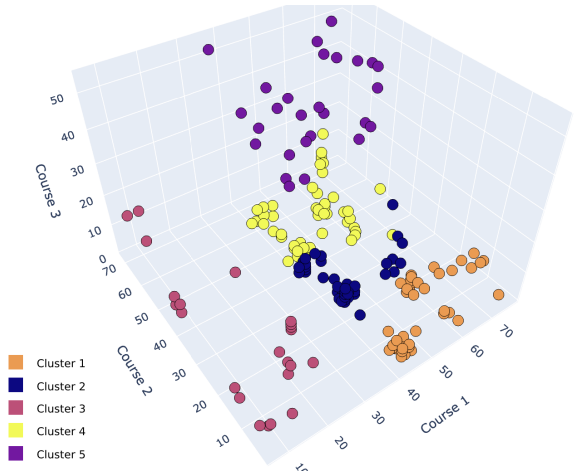

FIGURE 4. 3D plotting k-means clustering result.

dividing the number of ontologies found in a cluster by the total number of ontologies. As can be seen, not all root ontologies were satisfied. In this instance, the case study contained a thesis title whose substance could not be mapped to a specific root ontology. For instance, students rarely choose the title of a thesis relevant to a network. In addition, most students were interested in titles associated with information systems. Nevertheless, each cluster has its own distinct set of traits. Clusters 1, 2, and 3 focused on the software engineering thesis topics. Cluster 1 applies software engineering to information system products such as personal health assistant applications, crowd-reporting applications, and hospital logistics information systems. Cluster 2 combines software engineering with the scientific theory of computation, computational algorithms, and intelligent systems, such as digital whiteboards, smart e-catering applications, and multiplayer game nonograms. Finally, Cluster 3 combines software engineering with aspects of human-computer interaction in the products produced, such as life simulation games, intuitive bowling game applications, and virtual reality physics simulation. The substance described by Cluster 3 was distinct from that described by the other clusters. Most of the titles in Cluster 3 focused on educational topics, video games, and the connection between humans and computers.

Cluster 4 predominantly covered information system ontology, with e-commerce websites for small businesses, e-government applications, and job recommendation information systems as examples. This cluster is more towards the title of software engineering, which is applied to

Information systems projects with examples of leaf ontologies, such as development frameworks, compilers, and software maintenance tools. Finally, cluster 5 focuses on computing methodologies that produce an information system output. For example, information systems for the logistical needs of victims of natural disasters, sentiment analysis, and decision support systems for purchasing goods. In addition, the majority of the systems in Cluster 5 belong to intelligent system topics. These systems include information retrieval, machine learning, and artificial intelligence.

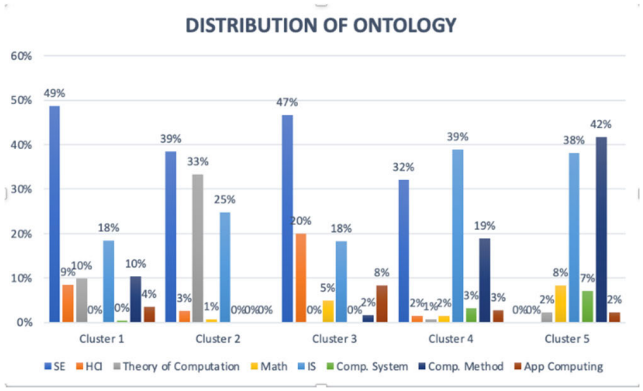

FIGURE 5. Distribution of ontology.

It is possible to draw the conclusion that given that cluster 2 is the largest cluster, most titles pertaining to information systems are gathered in this cluster with leaf ontologies, including enterprise computing, data management systems, and information system applications. The software engineering ontology dominated all clusters with a percentage of 35.32%, followed by the information system ontology with 27.37%. The remaining ontology was distributed evenly, with a theory of computation of 12.80% and applied computing at 11.70%. A curriculum team can use the results of this cluster analysis to evaluate and improve the curriculum. For instance, a study program can determine the research direction by examining the distribution of ontologies in clusters. Additionally, the cluster analysis results can serve as a reference for developing prediction, decision support, and thesis recommendation systems.

Fig 4 shows a 3D plot representing the k-means clustering outcome. The plot illustrates the separation of the 300 data points into five distinct clusters. Each axis corresponds to the encoding of supporting courses associated with each data point, and each data point represents a thesis. As shown in Fig 4, the distribution of data points is dependent on the encoding of the three supporting courses outlined in Table 4. For instance, data points 0-10 belong to the Software Engineering ontology, whereas data points 53-63 belong to the computing methodology ontology. Based on a visual inspection of the 3D plot, the dataset exhibits non-uniform clustering with varied shapes. Consequently, density-based algorithms yield suboptimal results.

Fig 5 shows our research in a visually appealing manner, presenting the scientific content of various thesis titles. Through our analysis, we identified the top five ontologies: software engineering, information systems, human-computer interaction, computing methodologies, and applied computing. By counting the ontologies in each cluster and calculating the percentage distribution, we highlighted the predominant ontology and its distribution within the dataset. Each cluster is also associated with a course, such as ''Intelligent Information Retrieval'' in Cluster 0, ''Applied Database'' in Cluster 1, ''Software Engineering'' in Cluster 2, ''Enterprise System Implementation'' in Cluster 3, and ''Human-Computer Interaction'' in Cluster 4. This information can be used to develop a recommendation system for thesis topics and titles, considering the relevance of courses to students' abilities.

TABLE 8. Correlation analysis summary.

Cluster #	Σ	μD (months)	μ GPA	ρ (GPA, D)	ρ (C, D)
1	30	6.68	3.63	-0.3242469726	-0.3246881753
2	90	9.02	3.23	-0.3865122321	-0.3661417629
3	70	9	3.36	-0.5156894356	-0.4460672125
4	66	8.03	3.55	-0.528283729	-0.420287703
5	46	9.28	3.31	-0.4853516519	-0.4037355567

We examined how GPA and the duration of a thesis are linked, as well as how the average grades of thesis-supporting courses and the duration of a thesis are related. We took each thesis data point and charted the students' GPA and the average grades of thesis-supporting courses. We used a Pearson correlation analysis to determine the correlation between GPA, the average grades of thesis-supporting courses, and the duration of the thesis. Using a scatter plot, Fig 6 illustrates the distribution of data points based on completion time, GPA, and average grades of thesis-supporting courses. The trend for completion time demonstrates a connection between GPA, the average grades of thesis-supporting courses, and the duration of the thesis. The higher the GPA and average grades of thesis-supporting courses, the shorter the duration of the thesis.

The results of the Pearson's correlation analysis are presented in Table 8. The Σ symbol represents the total number of data points in a cluster, μD denotes the average time (in months) taken to complete a thesis for a specific cluster, and μGPA signifies the average GPA for the same cluster. The symbol ρ(GPA, D) shows the correlation test outcome between GPA and the duration of thesis completion in a specific cluster, whereas ρ(C, D) represents the correlation test outcome between the average grades of thesis-supporting courses and the duration of thesis completion. The correlation values suggest a moderate correlation between GPA and grades in supporting courses with the length of thesis completion. Negative values indicate an inverse correlation, meaning that lower GPA values correspond to longer thesis completion times for students, whereas higher GPA values correspond to faster thesis completion times. This trend was also observed in the correlation between grades in the average grades of thesis-supporting courses and thesis completion duration.

Analysis of these clusters revealed that Cluster 1 had the shortest average completion time of 6.68 months. This grouping comprised 30 theses that focused on software engineering for information system products. Interestingly, students with higher GPAs completed their theses faster in this cluster, indicating a negative correlation of -0.3242 between GPA and thesis duration. In contrast, cluster 5 had the longest average completion time of 9.28 months. This cluster consists of 46 theses that focus on intelligent systems. Here, a moderately negative correlation of -0.4854 between GPA

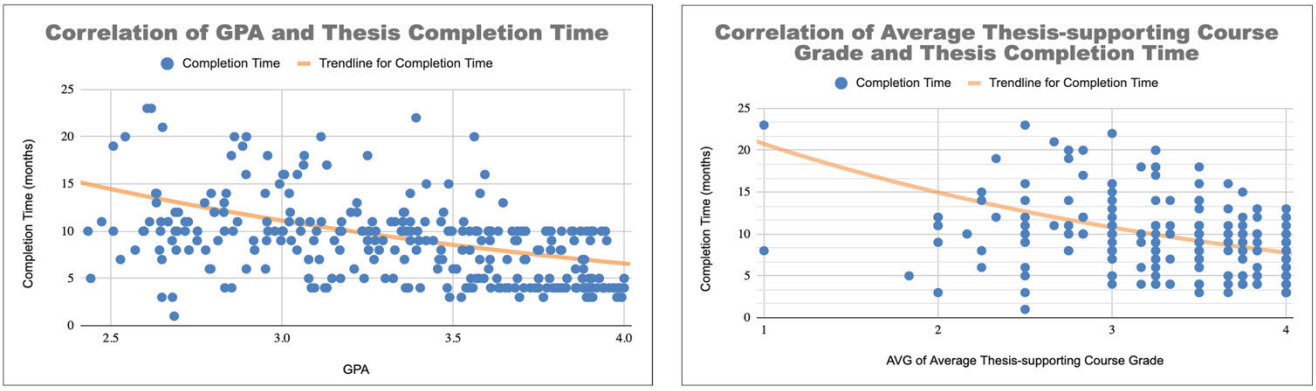

FIGURE 6. Scatter plot diagram of correlation between GPA, supporting course grade and thesis completion time.

and thesis duration was observed, signifying that students with higher GPAs completed their theses more quickly in this cluster, paralleling the results seen in Clusters 1 and 2. Furthermore, we observed that Cluster 5, which took the longest to complete, featured considerable ontological content pertaining to mathematics, as depicted in Fig 5. Mathematics is a significant domain in computer science programs, and further research is required to determine whether proficiency in and comprehension of mathematics contribute to the smooth progression of thesis work. These findings underscore the importance of selecting a suitable thesis topic to ensure its timely completion.

Regarding certain clusters, Cluster 1 indicates that GPA and average grades in thesis-supporting courses have a minimal effect on thesis completion time within this cluster. However, Cluster 4 exhibited the highest correlation between GPA and thesis duration, emphasizing the significance of academic performance in expediting thesis completion. Similarly, Cluster 3 displayed the strongest correlation between average grades in thesis-supporting courses and thesis duration, highlighting the importance of good performance in these courses for timely thesis completion. Nevertheless, the moderate correlation results suggest that factors beyond GPA and average grades in supporting courses contribute to thesis completion duration. Further studies are necessary to explore additional influences, such as student motivation, on thesis work. This investigation could delve into areas such as procrastination, confidence levels, and student autonomy in selecting appropriate topics. Moreover, the impact of supervisory guidance styles, supervisor reputation, and alignment between student-selected topics and advisors' expertise should be scrutinized. Finally, the influence of academic abilities reflected in students' academic transcripts and the number of repeated courses should be investigated. These findings provide valuable insights for future studies. We can examine the differences in correlations between clusters to better understand how diverse academic programs prepare students for their theses. This information can guide us in evaluating and improving our academic programs to ensure that they adequately equip students with their research endeavors.

However, it is essential to note that clustering results may vary when using datasets from other universities. Our study shows the effectiveness of k-means clustering in mapping each study program's knowledge composition and distribution patterns. This provides valuable insights for future research in higher education and aids in developing topic-recommendation procedures. Our study can serve as a benchmark for future research in this field.

V. CONCLUSION

In this study, we investigated the EDM dataset to discover concealed data and operational patterns among 300 titles from a thesis course at the University of Surabaya. Students must apply the skills and knowledge gained during their education by working on the thesis course. During their work, as part of their requirements, students also demonstrated the ability to think critically, creatively, and independently while receiving guidance from a supervisor or mentor. Regrettably, delays in completing undergraduate theses are common in universities. This matter is concerning because delays in completing the thesis might also negatively affect students' grades and the institution's accreditation. One of the most common reasons is that students select thesis topics that are not well suited to their competencies. Therefore, a suitable thesis topic based on students' academic records could solve this thesis delay problem.

Second, we investigated clustering techniques that are practical and efficient for the problem. We prepared a dataset extracted from past undergraduate theses and annotated it using three supporting courses. Based on a comparison of the clustering techniques, we conclude that k-means is an effective and efficient algorithm for this dataset. The clustering process produces five clusters. Furthermore, we conclude that applying the k-means technique to other university datasets is possible and should deliver different insights and cluster patterns. Through a series of experiments and processes, this study significantly contributes to the understanding and evaluation of the learning outcomes of study programs defined in the curriculum, following the design and implementation of thesis topics. Similarly, the clustering results are essential building blocks for future work. Eventually, this study will

benefit higher education as a reference for formulating a study program research roadmap.

Thus, there is room for improvement in future studies. This includes data preparation step outcomes that depend highly on the expert's judgment, which means that annotation mistakes are still possible. Another annotation method that should be considered is crowdsourced annotation, which is more cost effective. In this case, we consider employing a group of lecturers and students on specific topics of expertise as crowdsourced in labeling our dataset. Second, the manual annotation process conducted by an expert can benefit significantly if a student's thesis supervisor is involved. This supervisor should be more familiar with the factual content of the thesis than the independent experts. Therefore, it can reduce mislabeling errors during the annotation process. Finally, our k-means algorithm uses a nonunweighted dataset. This means that all three features of the supporting course have the same proportion and influence in supporting the content of the student undergraduate thesis. For example, some courses may have a dominant influence on thesis title compared to other courses. By implementing weighted clustering, we can annotate additional information regarding the weights of the features in each undergraduate thesis.

Research on EDM data has yielded significant results, particularly in student thesis clustering. This study has the potential to improve and maintain better outcomes. These findings can be used to develop a tailor-made method for suggesting topics for undergraduate theses based on individual preferences and interests. This research can also be applied to other fields of study using a standard ontology that aligns with the domain knowledge of those fields. Moreover, the research indicates that factors beyond GPA and average grades in supporting courses affect the time taken to complete a thesis. Future studies should explore other influences such as student motivation, procrastination, confidence levels, and autonomy in topic selection. Additionally, it is crucial to investigate the impact of supervisory guidance style, supervisor reputation, student-topic alignment, academic ability reflected in academic transcripts, and number of repeated courses.

REFERENCES

[1] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," *Exp. Syst. Appl.*, vol. 33, no. 1, pp. 135–146, Jul. 2007, doi: 10.1016/j.eswa.2006.04.005.

[2] I. O. Pappas, M. N. Giannakos, L. Jaccheri, and D. G. Sampson, "Assessing student behavior in computer science education with an fsQCA approach: The role of gains and barriers," *ACM Trans. Comput. Educ.*, vol. 17, no. 2, pp. 1–23, Jun. 2017, doi: 10.1145/3036399.

[3] M. Durairaj and C. Vijitha, "Educational data mining for prediction of student performance using clustering algorithms," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 5987–5991, 2014.

[4] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Student engagement level in an e-learning environment: Clustering using K-means," *Amer. J. Distance Educ.*, vol. 34, no. 2, pp. 137–156, Apr. 2020, doi: 10.1080/08923647.2020.1696140.

[5] B. Rawat and S. K. Dwivedi, "Discovering learners' characteristics through cluster analysis for recommendation of courses in e-learning environment," *Int. J. Inf. Commun. Technol. Educ.*, vol. 15, no. 1, pp. 42–66, Jan. 2019, doi: 10.4018/ijicte.2019010104.

[6] V. Efrati, C. Limongelli, and F. Sciarrone, "A data mining approach to the analysis of students' learning styles in an e-learning community: A case study," in *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge*. Cham, Switzerland: Springer, 2014, pp. 289–300.

[7] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," *IEEE Trans. Syst., Man, Cybern., C*, vol. 40, no. 6, pp. 601–618, Nov. 2010, doi: 10.1109/TSMCC.2010.2053532.

[8] L. Khanna, S. N. Singh, and M. Alam, "Educational data mining and its role in determining factors affecting students academic performance: A systematic review," in *Proc. 1st India Int. Conf. Inf. Process. (IICIP)*, Aug. 2016, pp. 1–7, doi: 10.1109/IICIP.2016.7975354.

[9] K. T. S. Kasthuriarachchi, S. R. Liyanage, and C. M. Bhatt, "A data mining approach to identify the factors affecting the academic success of tertiary students in Sri Lanka," in *Software Data Engineering for Network eLearning Environments*, S. Caballé and J. Conesa, Eds. Cham, Switzerland: Springer, 2018, pp. 179–197, doi: 10.1007/978-3-319-68318-8_9.

[10] H. Jeong and G. Biswas, "Mining student behavior models in learning-by-teaching environments," in *Proc. Educ. Data Mining*, Canada, 2008, pp. 127–136. [Online]. Available: https://www.educationaldatamining.org

[11] S. Kausar, X. Huahu, I. Hussain, Z. Wenhao, and M. Zahid, "Integration of data mining clustering approach in the personalized e-learning system," *IEEE Access*, vol. 6, pp. 72724–72734, 2018, doi: 10.1109/ACCESS.2018.2882240.

[12] F. Del Bonifro, M. Gabbrielli, G. Lisanti, and S. P. Zingaro, "Student dropout prediction," in *Artificial Intelligence in Education*. Cham, Switzerland: Springer, 2020, pp. 129–140.

[13] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques," *Comput. Educ.*, vol. 53, no. 3, pp. 950–965, Nov. 2009, doi: 10.1016/j.compedu.2009.05.010.

[14] F. Yang and F. W. B. Li, "Study on student performance estimation, student progress analysis, and student potential prediction based on data mining," *Comput. Educ.*, vol. 123, pp. 97–108, Aug. 2018, doi: 10.1016/j.compedu.2018.04.006.

[15] N. Tomasevic, N. Gvozdenovic, and S. Vranes, "An overview and comparison of supervised data mining techniques for student exam performance prediction," *Comput. Educ.*, vol. 143, Jan. 2020, Art. no. 103676, doi: 10.1016/j.compedu.2019.103676.

[16] S. M. Razaulla, M. Pasha, and M. U. Farooq, "Integration of machine learning in education: Challenges, issues and trends," in *Machine Learning and Internet of Things for Societal Issues*, C. Satyanarayana, X.-Z. Gao, C.-Y. Ting, and N. B. Muppalaneni, Eds. Singapore: Springer, 2022, pp. 23–34, doi: 10.1007/978-981-16-5090-1_2.

[17] (2022). *International Educational Data Mining Society*. Accessed: Mar. 9, 2022. [Online]. Available: https://educationaldatamining.org/

[18] A. Dutt and M. A. Ismail, "Logical review on educational data mining," *Int. J. Comput. Commun. Netw.*, vol. 9, no. 3, pp. 39–42, 2020, doi: 10.30534/ijccn/2020/01932019.

[19] M. Munoz-Organero, P. J. Munoz-Merino, and C. D. Kloos, "Student behavior and interaction patterns with an LMS as motivation predictors in e-learning settings," *IEEE Trans. Educ.*, vol. 53, no. 3, pp. 463–470, Aug. 2010.

[20] C. Su, "Designing and developing a novel hybrid adaptive learning path recommendation system (ALPRS) for gamification mathematics geometry course," *EURASIA J. Math., Sci. Technol. Educ.*, vol. 13, no. 6, pp. 2275–2298, Apr. 2017, doi: 10.12973/eurasia.2017.01225a.

[21] S. Maldonado, J. Miranda, D. Olaya, J. Vásquez, and W. Verbeke, "Redefining profit metrics for boosting student retention in higher education," *Decis. Support Syst.*, vol. 143, Apr. 2021, Art. no. 113493, doi: 10.1016/j.dss.2021.113493.

[22] J. Tondeur, S. K. Howard, and J. Yang, "One-size does not fit all: Towards an adaptive model to develop preservice teachers' digital competencies," *Comput. Hum. Behav.*, vol. 116, Mar. 2021, Art. no. 106659, doi: 10.1016/j.chb.2020.106659.

[23] S. B. Aher and L. M. R. J. Lobo, "Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data," *Knowl.-Based Syst.*, vol. 51, pp. 1–14, Oct. 2013, doi: 10.1016/j.knosys.2013.04.015.

[24] S. A. Priyambada, M. Er, B. N. Yahya, and T. Usagawa, "Profile-based cluster evolution analysis: Identification of migration patterns for understanding student learning behavior," *IEEE Access*, vol. 9, pp. 101718–101728, 2021, doi: 10.1109/ACCESS.2021.3095958.

[25] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, Dec. 2017, doi: 10.1016/j.neucom.2017.06.053.
[26] K. H. Tie, A. Senawi, and Z. L. Chuan, "An observation of different clustering algorithms and clustering evaluation criteria for a feature selection based on linear discriminant analysis," in *Enabling Industry 4.0 through Advances in Mechatronics*. Singapore: Springer, 2022, pp. 497–505.
[27] M. Mughnyanti, S. Efendi, and M. Zarlis, "Analysis of determining centroid clustering X-means algorithm with Davies-Bouldin index evaluation," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 725, no. 1, 2020, Art. no. 012128, doi: 10.1088/1757-899X/725/1/012128.
[28] R. Romaniuc and C. Bazart, "Intrinsic and extrinsic motivation," in *Encyclopedia of Law and Economics*. New York, NY, USA: Springer, 2015, pp. 1–4, doi: 10.1007/978-1-4614-7883-6_270-1.
[29] A. Petersen, M. Craig, J. Campbell, and A. Tafliovich, "Revisiting why students drop CS1," in *Proc. 16th Koli Calling Int. Conf. Comput. Educ. Res.*, Nov. 2016, pp. 71–80, doi: 10.1145/2999541.2999552.
[30] J. Nouri, K. Larsson, and M. Saqr, "Identifying factors for master thesis completion and non-completion through learning analytics and machine learning," in *Transforming Learning with Meaningful Technologies* (Lecture Notes in Computer Science), vol. 11722. Berlin, Germany: Springer, 2019, doi: 10.1007/978-3-030-29736-7_3.
[31] T. O'Donoghue and M. Rabin, "Procrastination on long-term projects," *J. Econ. Behav. Org.*, vol. 66, no. 2, pp. 161–175, May 2008, doi: 10.1016/j.jebo.2006.05.005.
[32] E. Irrazábal, M. A. Mascheroni, C. Greiner, and G. Dapozo, "Procrastination at the conclusion of the master's thesis: Results from a survey on computer science students in northeast Argentina," in *Proc. 43rd Latin Amer. Comput. Conf. (CLEI)*, Sep. 2017, pp. 1–6, doi: 10.1109/CLEI.2017.8226391.
[33] E. H. Seo, "The relationships among procrastination, flow, and academic achievement," *Social Behav. Personality, Int. J.*, vol. 39, no. 2, pp. 209–217, Mar. 2011, doi: 10.2224/sbp.2011.39.2.209.
[34] A. Andre, N. Suciati, and H. Fabroyir, "Expert annotation tools for labeling student capstone project based on ACM CCS ontology," in *Proc. 11th Electr. Power, Electron., Commun., Controls Informat. Seminar (EECCIS)*, Aug. 2022, pp. 345–350, doi: 10.1109/EECCIS54468.2022.9902931.
[35] C. Lipson, *How to Write a BA Thesis: A Practical Guide From Your First Ideas to Your Finished Paper*. Chicago, IL, USA: Univ. Chicago Press, 2018.
[36] Z. Liu, J. Liu, X. Xiao, H. Yuan, X. Li, J. Chang, and C. Zheng, "Segmentation of white blood cells through nucleus mark watershed operations and mean shift clustering," *Sensors*, vol. 15, no. 9, pp. 22561–22586, Sep. 2015, doi: 10.3390/s150922561.
[37] A. Karami and R. Johansson, "Choosing DBSCAN parameters automatically using differential evolution," *Int. J. Comput. Appl.*, vol. 91, no. 7, pp. 1–11, Apr. 2014, doi: 10.5120/15890-5059.
[38] Z. Falahiazar, A. Bagheri, and M. Reshadi, "Determining the parameters of DBSCAN automatically using the multi-objective genetic algorithm," *J. Inf. Sci. Eng.*, vol. 37, no. 1, pp. 157–183, 2021, doi: 10.6688/JISE.202101_37(1).0011.
[39] N. Rahmah and I. S. Sitanggang, "Determination of optimal epsilon (eps) value on DBSCAN algorithm to clustering data on peatland hotspots in Sumatra," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 31, no. 1, 2016, Art. no. 012012, doi: 10.1088/1755-1315/31/1/012012.
[40] B. Lorbeer, A. Kosareva, B. Deva, D. Softić, P. Ruppel, and A. Küpper, "A-BIRCH: Automatic threshold estimation for the BIRCH clustering algorithm," in *Advances in Big Data*, vol. 529. Cham, Switzerland: Springer, Apr. 2018, pp. 169–178, doi: 10.1007/978-3-319-47898-2_18.

ANDRE received the bachelor's degree in computer science from Universitas Surabaya, in 2006, and the M.Sc. degree in digital media technology from Nanyang Technology University, in 2012. He is currently pursuing the Ph.D. degree. He has been a Lecturer with Universitas Surabaya, since 2007. His research interests include game design and development, mobile development, XR development, and human–computer interface.

NANIK SUCIATI (Member, IEEE) received the master's degree in computer science from the University of Indonesia, in 1998, and the Dr.Eng. degree in information engineering from Hiroshima University, in 2010. She is currently an Associate Professor with the Department of Informatics, Institut Teknologi Sepuluh Nopember. She has published more than 50 journal articles and conference papers on computer science. Her research interests include computer vision, computer graphics, and artificial intelligence.

HADZIQ FABROYIR (Member, IEEE) received the Doctor of Computer Science and Information Engineering degree from the National Taiwan University of Science and Technology. In 2020, he joined the Department of Informatics, Institut Teknologi Sepuluh Nopember, as an Assistant Professor. His research interests include human–computer interaction focusing on virtual navigation and extended reality.

ERIC PARDEDE (Senior Member, IEEE) received the master's degree in information technology and the Ph.D. degree in computer science from La Trobe University, Melbourne, Australia. He is currently an Associate Professor with La Trobe University. He has published more than 150 publications in international journals, conference proceedings, and books. His research interests include data analytics, IT education, and entrepreneurship.