A Data-Driven Approach for Engineering Degree Programme Review Based on Graph Theory

Chao Shu
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, UK
chao.shu@qmul.ac.uk

Yue Chen
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, UK
yue.chen@qmul.ac.uk

Kok Keong Chai
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, UK
michael.chai@qmul.ac.uk

***Abstract*—This research full paper proposes a novel data-driven approach for programme review that leverages module assessment data in an undergraduate engineering degree programme and graph theory concepts. The approach involves constructing a curriculum correlation graph, where nodes represent modules and edge weights are determined by correlation coefficients between assessment results of all modules in the engineering programme. Based on the curriculum correlation graph, graph-theoretic techniques and metrics, such as the minimum spanning tree, clustering coefficients and centrality measures, are employed to perform quantitative analyses, which evaluate the coherence of the programme's curriculum delivery. Furthermore, the approach facilitates a quantitative evaluation of the alignment between the programme's intended curriculum structure, as encapsulated in the designed curriculum graph, and its actual delivery, represented by the curriculum correlation graph. By comparing centrality measures between these two graphs, the approach highlights areas where the programme's curriculum delivery may deviate from its original design expectations, allowing targeted interventions to address potential misalignments. The proposed approach is applied to a UK-China transnational education undergraduate engineering degree programme. The analysis results demonstrate the effectiveness of the proposed data-driven approach in providing comprehensive and quantitative insights into the programme's curriculum design and delivery. By leveraging the power of graph theory and data analysis techniques, this approach offers a valuable tool for programme review, enabling programme teams in higher education institutions to identify both strengths and potential discrepancies in the alignment between a programme's curriculum delivery and its original design expectations, so that informed decision and targeted efforts can be made towards continuous improvement and enhancement of the academic degree programme.**

***Index Terms*—Programme evaluation, Engineering curriculum, data analytics, graph theory**

I. Introduction

Programme review in higher education, specifically for engineering programmes, is a crucial process that ensures the relevance and quality of the curriculum. It aims at improving the effectiveness of the programme by identifying areas of strength and areas that require improvement [1], [2]. The commonly used approaches include surveys, interviews, and data analysis [3]. Evaluating the effectiveness of teaching and assessment is an important aspect of degree programme review. Conventionally, this process is conducted by comparing individual module results based on the exam board data. However, it is difficult to gather a holistic view of the entire degree programme.

Graph theory emerges as a transformative approach for presenting and analysing curricula. A graph-theoretic approach to analysing curriculum structure with a focus on assessment placement design is described in [4]. It represents the curriculum as a graph and uses quantitative metrics to identify optimal courses for assessment components. In [5], a systematic approach based on graph theory is used to analyse the structure of engineering programmes. Additionally, in order to measure the role that the structure of a curriculum plays in student academic success, several metrics are developed to evaluate curricular efficiency in [6]. However, these approaches only rely on prerequisite relationships between modules, aimed at helping the curriculum design process rather than the programme evaluation and quality enhancement.

In this research paper, we propose a data-driven graph-theoretic approach for engineering programme review that aims to address the following research questions:

1) How to identify module(s) that may need further review to maintain curriculum coherence?
2) How do different modules correlate with others in terms of their category/nature based on their assessment results, and whether the correlation is consistent with the design of the programme?

In the proposed approach, a curriculum correlation graph is constructed for a programme, where nodes represent modules, and Pearson's correlation coefficients between the assessment results of all modules within the programme are used as the edge weights. Differing from the curriculum graph in which the edges only represent the prerequisite relationship between modules decided at the programme/curriculum design stage, this curriculum correlation graph incorporates the results of the teaching and assessment delivery throughout all development years in the programme, reflecting the actual delivery of the programme's curriculum. After the graph is constructed, quantitative analyses based on graph theory are performed. The clustering coefficient is used as a quantitative measure to assess the coherence of the curriculum delivery in terms of the assessment results, identify cohesive clusters of modules and potential outliers or areas requiring further examination. The minimum spanning tree from the curriculum correlation graph using negated edge weights is derived to reveal the strongest connections among modules in the actual operation of the programme. Furthermore, by analysing centrality measures, such as degree centrality, betweenness centrality, and eigenvector centrality in the minimum spanning tree of the curriculum correlation graph and comparing them with those derived from the curriculum graph based on curriculum prerequisite design, we can evaluate the alignment between the programme's curriculum delivery and its original design expectations.

The proposed approach is applied to an accredited UK-China transnational education (TNE) undergraduate engineering degree programme. This programme is jointly developed and delivered by two universities, highlighting the importance of ensuring a seamless alignment between the design and delivery of the programme's curriculum through the programme review process. The analysis results demonstrate that the data-driven approach can effectively evaluate the coherence of the programme curriculum delivery, identify anomalies and reveal possible areas for improvement in the programme. It enables comprehensive quantitative analyses that unveil valuable insights into both the programme's curriculum design and its actual delivery. Compared to conventional analysis, this method provides a holistic view, and the insights revealed can facilitate continuous programme enhancement and contribute to a more cohesive, well-integrated curriculum design, ultimately enhancing the learning experience and outcomes for students.

The remainder of this paper is organised as follows. We start with building a curriculum graph based on the designed prerequisite relationships in the programme's curriculum as the evaluation baseline in Section II. The proposed data-driven approach for programme review is elaborated in Section III, followed by the demonstration of the approach with data analysis results in Section IV. The conclusions are drawn in Section V.

II. Curriculum Graph for Curriculum Design

The curriculum of an undergraduate programme can be naturally represented by a graph based on the designed prerequisite relationship between modules, where modules are represented as nodes and prerequisite relationships are represented as edges. In most undergraduate programmes, particularly in engineering disciplines, curricula often follow a structured sequence where fundamental concepts and skills (e.g., maths and programming) are introduced in early modules, and more advanced or specialised topics (e.g., subject-specific software/hardware/system engineering) build upon these foundations. This sequential nature of learning is well-represented by a directed acyclic graph (DAG), where the direction of edges indicates the flow of prerequisite knowledge from one module to another.

This DAG representation is useful for organising modules at the stage of designing a programme's curriculum and helpful when illustrating the designed curriculum to the relevant stakeholders. Given a designed curriculum C, it can be represented by a DAG $G_C = (V, E)$, where the nodes $v_1, v_2, ..., v_n \in V$ represent the modules in the curriculum, and there is a directed edge $(v_i, v_j) \in E$ from node i to node j if module i is the prerequisite of module j.

We applied this technique and constructed the curriculum graph for an engineering programme in a UK-China transnational education (TNE) collaboration. The curriculum graph, depicted in Fig. 1, represents the 4-year undergraduate programme spanning over 8 semesters. The graph is arranged chronologically from left to right, starting with Semester 1 and ending with Semester 7. It is noteworthy that students do not have any taught modules during the second semester of their final year as they focus on their final-year projects. The modules are divided into several groups. English, maths, and programming modules are modules that provide fundamental knowledge and skills. The rest of the modules in the curriculum are subject-specific modules and are further divided into different categories, such as subject introduction modules, software, hardware, and system engineering modules, as well as a practice module in Semester 7.

The curriculum graph in Fig. 1 reveals an intentionally designed structure in the module distribution across the academic years. The number of modules increases from Year 1 to Year 2, providing a gradual progression in the workload. Subsequently, the module count decreases from Year 2 to Year 4, allowing students to focus more on advanced and specialised topics as they progress through the programme. Furthermore, the graph clearly illustrates that foundational modules are strategically placed in the first two years, laying the groundwork for more advanced concepts. Notably, the mathematics and programming modules serve as prerequisites for numerous modules in Years 2 and 3, underscoring their importance for subsequent modules in the engineering programme curriculum. Additionally, English modules are considered an essential part of the fundamental modules in TNE programmes since research shows a correlation between students' English language proficiency and their academic performance in a TNE programme [7].

The advantage of representing a curriculum as a graph lies in its ability to unveil various curriculum characteristics by simply altering the graph layout while preserving the inter-module relationships. Instead of organising the curriculum graph in chronological order, we can present it solely based on module connections, as shown in Fig. 2. It can be observed from Fig. 2 that the curriculum graph naturally segregates into distinct clusters. Interestingly, most subject-specific software, hardware, and system engineering modules form a cluster with all programming modules (the left-most cluster), while subject-specific theoretical and introductory modules form a cluster with all mathematical modules. This clustering pattern underscores the deliberate design choices inherent in the engineering programme's curriculum structure.

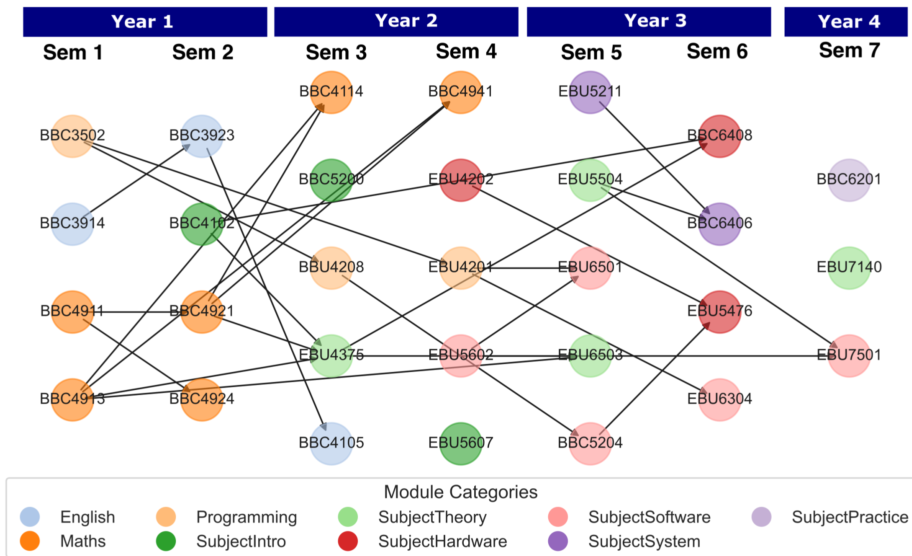


Fig. 1. The curriculum graph based on the designed prerequisite relationships between modules.

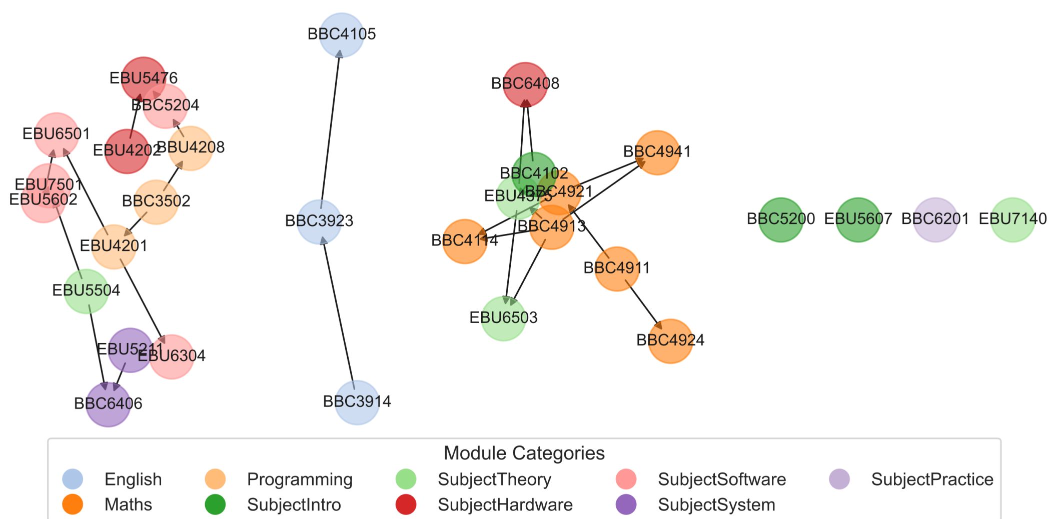


Fig. 2. The curriculum graph displayed in clusters.

Furthermore, by analysing various graph metrics of a curriculum graph, such as degrees of nodes, longest paths, centralities, etc., the structure of the curriculum graph can be evaluated, which can help improve the curriculum design [4]–[6].

However, it's crucial to note that since the curriculum graph is constructed solely on designed inter-module prerequisite relationships, it can not fully capture how the programme operates in practice. Therefore, while valuable for assessing structural integrity, the curriculum graph alone may not be sufficient to evaluate a programme's performance in practice and determine if the actual delivery aligns with the intended curriculum design objectives during the programme review.

III. Curriculum Correlation Graph for Programme Review

In order to reflect the actual operation of a programme with a curriculum C and quantitatively evaluate the programme's coherence and alignment with the curriculum design, we propose a new curriculum correlation graph, in which Pearson's correlation coefficients between the assessment results of all modules are used as the edge weights. The assumption that prerequisite relationships are reflected in correlation coefficients is based on the expectation that modules designed to build upon one another should show a degree of statistical dependency on student performance, given that the delivery and assessments can coherently differentiate students' competence in the aligned outcomes across modules.

A. Module Correlation Matrix

Firstly, module marks of all students over four development years in an engineering programme are collected. Suppose there are K modules in total over the four development years, and N students' academic records, including their marks of all K modules, are collected. Denoting student j's ($1 < j < N$) mark for module i ($1 < i < K$) as m_{ij}, the Pearson's correlation coefficient between the marks of all N students for module u and module v can be calculated as

$$r_{uv} = \frac{\sum_{j=1}^{N}(m_{uj} - \overline{m}_u)(m_{vj} - \overline{m}_v)}{\sqrt{\sum_{j=1}^{N}(m_{uj} - \overline{m}_u)^2}\sqrt{\sum_{j=1}^{N}(m_{vj} - \overline{m}_v)^2}} \tag{1}$$

where $\overline{m}_u$ and $\overline{m}_v$ are the average marks of module u and module v.

After Pearson's correlation coefficients between every pair of modules are calculated, the module correlation matrix can be constructed as

$$\mathbf{M} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1K} \\ r_{21} & r_{22} & \cdots & r_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ r_{K1} & r_{K2} & \cdots & r_{KK} \end{bmatrix} \tag{2}$$

B. Curriculum Correlation Graph

By using the module correlation matrix as the adjacency matrix, we create the curriculum correlation graph, which is a complete, un-directed graph $G_C^r = (V, E^r)$ (a complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge), where the nodes $v_1, v_2, ..., v_K \in V$ represent the modules in the programme curriculum, and the weight of the edge $(v_i, v_j) \in E$ between node v_i to node v_j denoted as w_{ij} is the Pearson correlation coefficient r_{ij} between students' marks of module i and j obtained in Section III-A. It is worth noting that we deliberately set the diagonal of $\mathbf{M}$ to be 0 before we use it as the adjacency matrix $\mathbf{W}$ to construct the curriculum correlation graph to avoid the meaningless self-loop for each node in the curriculum correlation graph.

$$\mathbf{W} = \begin{bmatrix} 0 & r_{12} & \cdots & r_{1K} \\ r_{21} & 0 & \cdots & r_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ r_{K1} & r_{K2} & \cdots & 0 \end{bmatrix} \tag{3}$$

The curriculum correlation graph, constructed based on the assessment results of all modules within the programme, encapsulates valuable information about the programme's actual implementation. For modules within the curriculum that are designed to have prerequisite relationships or are closely related in terms of topics and intended learning outcomes (ILOs), relatively strong correlations in their assessment results are anticipated. By using Pearson's correlation coefficients between the assessment results of all modules as the weights of the edges in the graph, the curriculum correlation graph effectively reflects the actual connections between the modules based on their delivery. As a result, this curriculum correlation graph can be a valuable tool for quantitative analysis and evaluation of programme operation, offering insights into its alignment with initial curriculum design expectations.

C. Clustering Measures

After the correlation coefficient graph for the programme is constructed, we can use clustering measures for a weighted graph, such as the clustering coefficient, to analyse the delivery of the programme curriculum.

The clustering coefficient is a measure used in graph theory to quantify the degree to which nodes in a graph tend to cluster together based on their weighted connections, whose value ranges from 0 to 1, with 1 meaning a complete graph [8], and is defined as the geometric average of the subgraph edge weights [9]:

$$c_u = \frac{1}{deg(u)(deg(u)-1))}\sum_{vw}(\hat{w}_{uv}\hat{w}_{uw}\hat{w}_{vw})^{1/3}. \tag{4}$$

where the edge weights $\hat{w}_{uv}$ are normalized by the maximum weight in the network $\hat{w}_{uv} = w_{uv}/\max(w)$. The value of c_u is assigned to 0 if $deg(u) < 2$.

When applied to our curriculum correlation graph, where edge weights represent the correlation coefficients between modules' assessment results, the clustering coefficient for a specific node (module) measures how well the module is connected to other modules in the programme curriculum according to their assessment results. Given that module assessment results ideally evaluate students' comprehension of the ILOs and also their learning abilities in the subject area, a strong correlation in assessment results is anticipated among closely related modules, leading to high weights for the connections between them in the curriculum correlation graph. This pattern emerges due to the expectation that students who demonstrate proficiency in one module's assessments are likely to perform well in assessments for modules covering related concepts, skills, or subject areas. Consequently, a high clustering coefficient for a particular module signifies that it tends to form tight clusters with other related modules, indicating that the assessment of that particular module is well-designed and well-aligned with other related modules in the programme. It also suggests that the module plays an important role in the programme's curriculum. Additionally, high clustering coefficients for many modules in the curriculum coefficient graph imply that the programme curriculum is well-integrated and the delivery of the curriculum is coherent within the programme. Modules with low clustering coefficients, on the other hand, may be more isolated or loosely connected to other modules in the programme. These modules potentially require further review to ensure that their teaching and assessment strategies are appropriately aligned with the overall curriculum goals.

Therefore, by analysing the clustering coefficients of different modules in the curriculum correlation graph, programme review team can identify potential areas of strength or weakness in the curriculum's structure and coherence of the curriculum delivery.

D. Minimum Spanning Tree

In a weighted connected graph, the minimum spanning tree (MST) is a subset of edges that connects all the vertices (nodes) of the graph without forming any cycles while minimising the total weight of its edges. Essentially, it represents the most efficient or compact way of connecting all nodes within the graph based on the given edge weights.

In the context of the curriculum correlation graph, where edge weights represent the correlation coefficients between modules' assessment results, we obtain its MST by using the negated edge weights ($-\mathbf{W}$) before applying the MST algorithm. In this way, we effectively connect all modules in the programme's curriculum with the strongest correlations between them in terms of assessment results and reveal a representation of a backbone structure of the curriculum in terms of its actual delivery. Modules that appear as central nodes or hubs in the MST can be considered core modules whose teaching content and assessments are well-designed so that they serve as anchors for the programme's educational framework that correlate with different modules in the subject area.

Further analysis of the MST of the curriculum correlation graph based on graph metrics, such as centralities, can help programme review team find insights into the actual operation of the programme's curriculum and evaluate it against the curriculum design expectations.

E. Measures of Centrality

Centrality measures are important concepts in graph theory that quantify the relative importance or influence of nodes within a graph structure. Among the commonly used centrality measures, degree centrality, betweenness centrality, and eigenvector centrality, can also be applied to the analysis of curriculum graphs.

For a node v in a graph $G = (V, E)$, the degree centrality $C_D(v)$ is defined as the number of edges connected to v, i.e., the degree of v. Since we want to compare the graph metrics between the curriculum graph, which is the representation of the designed curriculum, and the curriculum correlation graph, which represents the actual performance of the curriculum delivery, we normalise the degree centrality values by dividing by the maximum possible degree in a simple graph in G.

$$C_D(v) = \frac{\deg(v)}{n-1} \tag{5}$$

where n is the number of nodes in G.

Degree centrality measures the number of edges connected to a node, indicating its immediate influence or connectivity within the graph. Nodes with a high degree of centrality are often considered central hubs within the network.

The betweenness centrality $C_B(v)$ is the sum of the fraction of all-pairs shortest paths that pass through v.

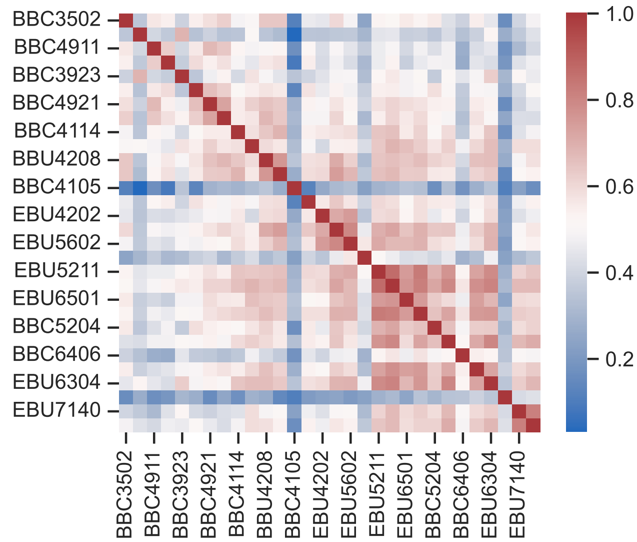


Fig. 3. The heatmap of the module correlation matrix.

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{6}$$

where σ_{st} is the total number of shortest paths from node s to node t, and $\sigma_{st}(v)$ is the number of those paths that pass through v.

Betweenness centrality quantifies the extent to which a node lies on the shortest paths between other pairs of nodes, acting as a bridge or bottleneck.

Eigenvector centrality considers not only the number of connections but also the centrality of the nodes to which a node is connected, reflecting the influence of a node within the network. The computation algorithm for Eigenvector centrality can be found in [10].

When it comes to the curriculum analysis, we derive these metrics from the curriculum graph and the MST of the curriculum correlation graph, respectively, and use them to quantitatively evaluate the characteristics of moudules in design and in practice, respectively.

Modules with high degree centrality are directly connected/associated to many other modules, suggesting that they are foundational or serve as prerequisites for a significant portion of the curriculum. Modules with high betweenness centrality might lie on critical connections between major clusters in the curriculum, facilitating the integration of knowledge across domains. Modules with high eigenvector centrality might signify core courses with strong connections to other central modules, forming the backbone of the curriculum.

Analysing these centrality measures allows for the identification of key modules that are central to the curriculum's structure and coherence and outliers that require further review, enabling targeted efforts to strengthen or improve these modules if necessary.

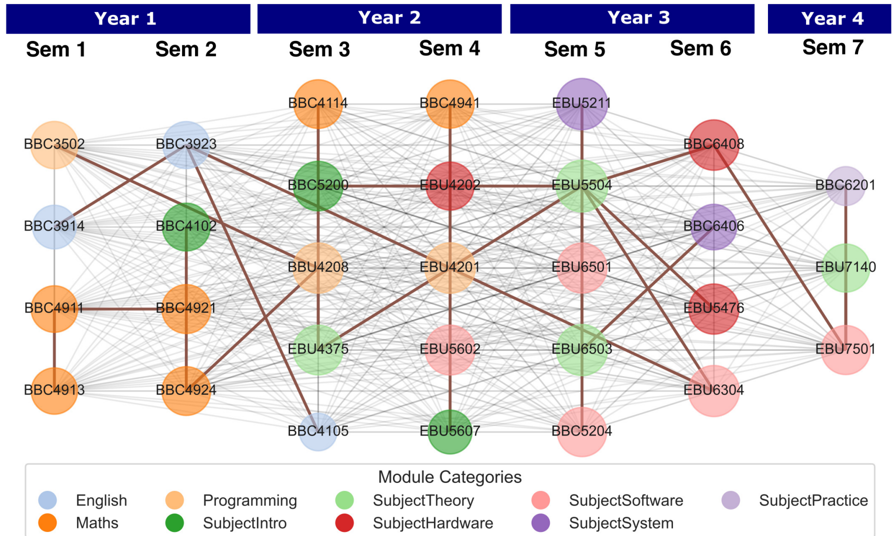


Fig. 4. The curriculum correlation graph for the engineering programme.

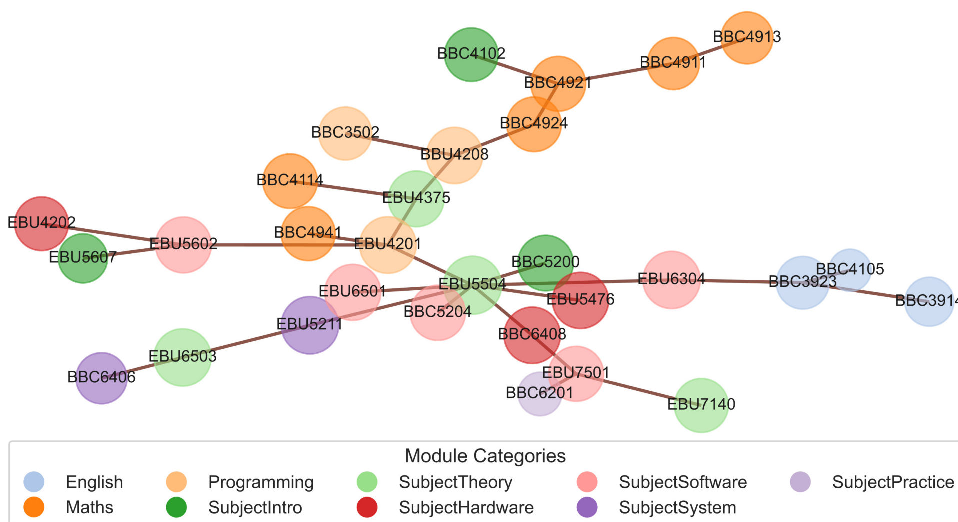


Fig. 5. The MST of the curriculum correlation graph for the engineering programme.

IV. Data Analysis Results

To demonstrate the effectiveness of our data-driven approach, we collected the comprehensive academic records of one cohort of students enrolled in the same engineering programme mentioned in Section II. To ensure a more accurate reflection of the programme's effectiveness in delivery, records of all retake students have been excluded from the analysis. The final dataset comprises the academic records of 171 students who initially enrolled in the programme in 2018 and completed their study in 2022. The collected data include all module marks obtained by every student throughout the four-year duration of the programme. Utilising this cohort as a representative sample, we applied the proposed curriculum correlation graph approach and derived the metrics elaborated in Section III, enabling a comprehensive analysis and evaluation of the programme's curriculum design and delivery.

Firstly, we derive the module correlation matrix $\mathbf{M}$ as described in Section III-A, and the heat map of $\mathbf{M}$ is presented in Fig. 3. Due to the limitation of the figure size, not all module codes are labelled in the figure. A preliminary observation from the heatmap suggests that assessment results from Year 3 modules exhibit relatively higher correlation coefficients than modules in other years, because the area corresponding to Year 3 modules displays a more intense colour gradient towards red, indicating stronger correlations between modules within this academic stage. Additionally, it is apparent that a few modules exhibit weak correlations with the majority of other modules.

Based on the module correlation matrix, the curriculum correlation graph is constructed as described in Section III-B and presented in Fig. 4, in which the size of each node is proportional to its clustering coefficient computed according to Section III-C. It is evident from the figure that the majority of modules exhibit good clustering coefficients within a close range, indicating good cohesion among them. A closer examination of the data reveals that the mean and standard deviation of the clustering coefficients across all modules in the programme are 0.56 and 0.07, respectively. This observation demonstrates a robust coherence in the teaching and assessment delivery throughout the programme. Furthermore, it can be observed that modules in Years 2 and 3 tend to exhibit slightly higher clustering coefficients compared to those in Years 1 and 4. This pattern is consistent with the preliminary observations made from the module correlation matrix/heat map and suggests a well-integrated design for the fundamental subject-specific modules delivered in Years 2 and 3. These modules, which form the core of the programme's curriculum, exhibit strong coherence, reflecting a thoughtful alignment of content and assessments within this critical stage of the programme.

On the other hand, this visual representation also enables programme review team to swiftly pinpoint modules with low clustering coefficients, indicating potential isolation or weak connectivity in terms of teaching and assessment delivery. Such modules may need further review to ensure that their teaching and assessment strategies are appropriately aligned with the overall programme specifications. In our analysis, BBC4105 and BBC6201 stand out as distinct outliers with notably low clustering coefficients. BBC4105 is an English module focusing on academic writing, while BBC6201 is a general engineering practice module. These findings suggest that the assessments within the programme may not heavily rely on students' academic writing skills, or that the academic writing training materials may not be closely related to the subject area of the engineering programme. Similarly, the design of the general engineering practice module BBC6201 may not be closely aligned with the ILOs of many subject-specific modules in the engineering programme. Both cases require further review to decide if the targeted effort should be made to improve the particular modules, or if certain aspects of the programme should be strengthened to enhance their integration and alignment.

In Fig. 4, the edges highlighted in brown show the MST of the curriculum correlation graph, which is also presented in Fig. 5 based on the tree structure to focus on the relationships between modules. In Fig. 5 we can easily identify two modules acting as central hubs in the MST: EBU4201 and EBU5504. EBU4201 is a module about Java programming, and EBU5504 is about networks and protocols, which interestingly are designed to be two of the core modules in this engineering programme.

Examining the prerequisite relationships between modules sheds light on their roles within the programme's curriculum design, and degree centrality serves as a valuable quantitative metric for this evaluation. To assess the programme's curriculum delivery against its design expectations, we compare the degree centralities in both the curriculum graph and the MST of the curriculum correlation graph, as illustrated in Fig. 6. In Fig. 6, the x-axis represents the degree centrality in the curriculum graph, reflecting curriculum design properties, while the y-axis depicts the degree centrality in the MST of the curriculum correlation graph, derived from actual curriculum delivery data. If the actual delivery of the programme's curriculum is closely aligned with the design expectations, all module points are expected to be distributed near the line $y = x$. It is clear from Fig. 6 that the majority of the modules are scattered around the $y = x$ line, which is a quick sign indicating a reasonable alignment of the programme's curriculum delivery with the curriculum design expectations. This initial observation is further supported by a detailed examination of Fig. 2 and Fig. 5, which reveals a good match between the connections in the MST of the curriculum correlation graph and the clusters of the curriculum graph. This finding reinforces the conclusion that the programme's curriculum delivery reasonably aligns with its design expectations, supported by the coherent relationship between curriculum connections and delivery outcomes.

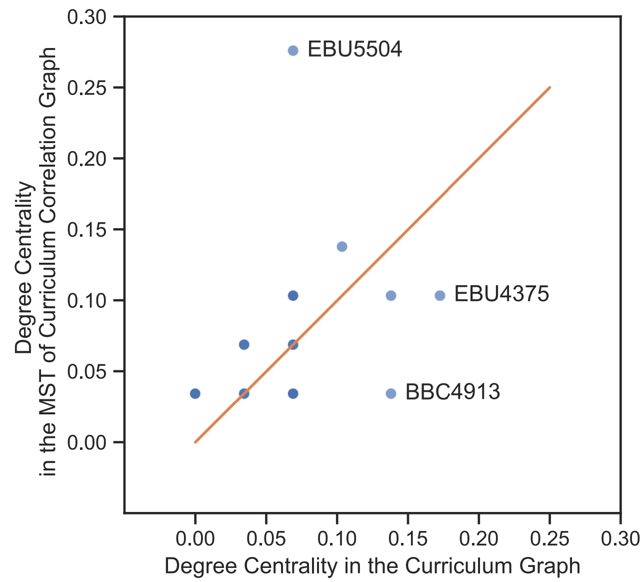
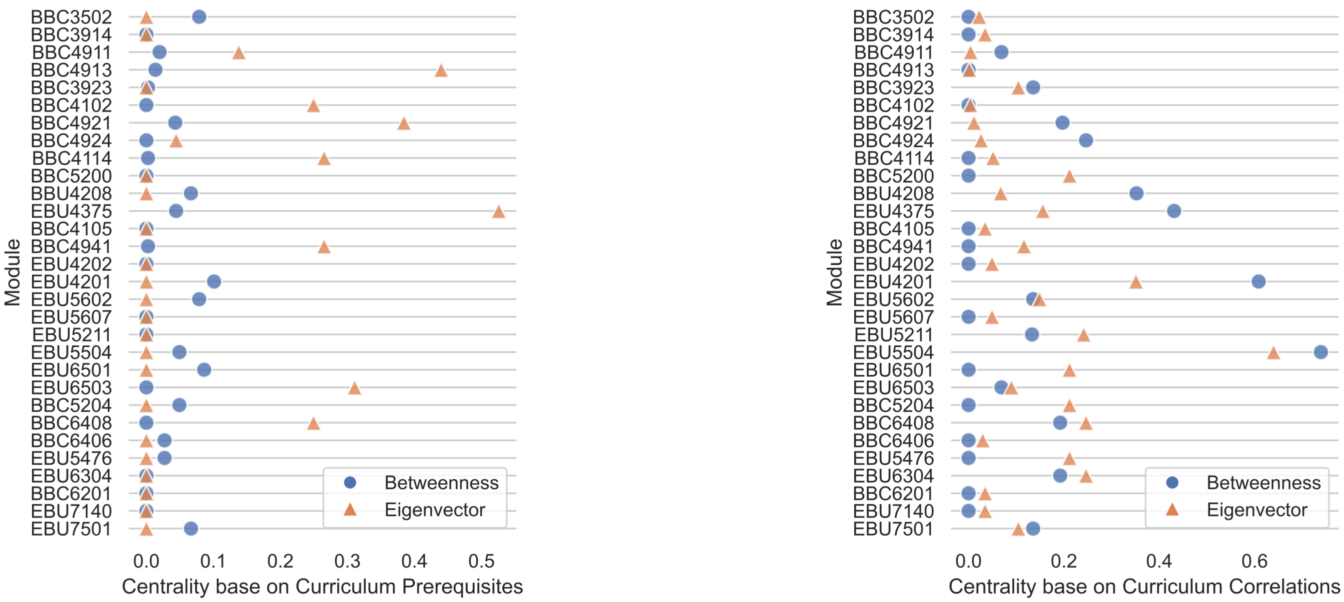


Fig. 6. The comparison of degree centralities in the curriculum graph and the MST of the curriculum correlation graph for all modules.

On the other hand, we can also identify a few outlier modules. The results reveal that EBU5504 shows more than expected connections with other modules, which is positive. However, for EBU4375 and BBC4913, which are fundamental modules, the results suggest their actual connections with other modules fall short of the design expectations, and a further review of the two modules may be required.

Fig. 7 depicts the betweenness and eigenvector centralities of all modules within the engineering programme derived from the curriculum graph and the MST of the curriculum correlation graph, respectively. These figures can be consid-



(a) (b)

Fig. 7. Betweenness and Eigenvector Centrality based on (a) Curriculum prerequisites and (b) Curriculum correlations.

ered as quantitative profiles for each module in terms of their association with other modules within the engineering programme based on the curriculum design and curriculum actual delivery results, respectively.

It can be observed that the betweenness centralities derived from the curriculum graph are all close to zero, which is reasonable, as we can see in Fig. 2 that the curriculum is designed to form several self-contained clusters and several independent modules. However, according to the results, the programme team may consider developing a module that can connect the two main clusters in the current curriculum: subject-specific application cluster with programming and subject-specific theory cluster with maths, which may be helpful for students in the programme to easily sense the big picture of the programme. The relatively high betweenness centralities derived from the curriculum correlation graph for some Year 2 and Year 3 modules imply there is a good alignment in the assessment standard across some modules in different categories and their well-designed assessments can differentiate students by their learning abilities, which leads to the stronger than designed betweenness centralities.

Interestingly, a noticeable disparity in eigenvector centralities can be observed between the curriculum's design and its delivery. While the designed curriculum exhibits high eigenvector centralities for Year 1 and Year 2 modules, this pattern does not align with the eigenvector centrality derived from the correlation of assessment results for these modules. Eigenvector centrality, which considers not only the number of connections but also the centrality of connected nodes, underscores the importance and influence of a node within the broader network. The group of foundational modules in Year 1 and Year 2 is designed to provide essential support for subsequent subject-specific modules, hence their high eigenvector centrality in the designed curriculum. However, these fundamental modules, such as mathematics, are often developed by teaching teams external to the School of Engineering, serving as common modules across various disciplines within the university. Consequently, their designed content and assessments may not seamlessly align with the ILOs of subsequent engineering modules in the programme. Only a subset of the knowledge imparted in these fundamental modules may be directly relevant or required for the engineering curriculum, causing the observed mismatch in eigenvector centralities between curriculum design and delivery. To address this issue, the programme team may consider collaborating with the mathematics teaching team to tailor these modules specifically for the engineering programme if necessary, ensuring better alignment with programme specifications and engineering student needs.

V. Conclusion

In this paper, we propose a novel data analysis approach for programme review based on programme assessment data and graph theory. We provide detailed explanations on the derivation and interpretation of quantitative metrics derived from graph theory, followed by a practical demonstration of our approach applied to an engineering programme. Through comprehensive and rigorous quantitative analyses, our methodology unveils valuable insights into both the curriculum design and delivery of the programme. The findings from our analyses underscore the efficacy and potential of our approach in evaluating and enhancing programme effectiveness.

REFERENCES

[1] E. El-Khawas, *Accountability and Quality Assurance: New Issues for Academic Inquiry*. Dordrecht: Springer Netherlands, 2007, pp. 23–37. [Online]. Available: https://doi.org/10.1007/978-1-4020-4012-2_3

[2] A. W. Frye and P. A. Hemmer, "Program evaluation models and related theories: Amee guide no. 67," *Medical Teacher*, vol. 34, pp. e288–e299, 5 2012.

[3] U. UK, "Framework for programme reviews: ensuring the value of courses," 1 2022. [Online]. Available: https://www.universitiesuk.ac.uk/sites/default/files/field/downloads/2022-01/uuk-framework-for-programme-reviews.pdf

[4] J. M. Lightfoot, "A graph-theoretic approach to improved curriculum structure and assessment placement," *Communications of the IIMA*, vol. 10, 2014.

[5] D. Varagnolo, S. Knorn, K. Staffas, E. Fjällström, and T. Wrigstad, "Graph-theoretic approaches and tools for quantitatively assessing curricula coherence," *European Journal of Engineering Education*, vol. 46, pp. 344–363, 5 2021.

[6] J. Wigdahl, G. L. Heileman, A. Slim, and C. T. Abdallah, "Curricular efficiency: What role does it play in student success?" 2014.

[7] C. Shu, Y. Chen, and K. K. Chai, "A data-driven analysis of the correlation between english language proficiency and academic performance in transnational education," in *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, 2023, pp. 1–5.

[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 6 1998.

[9] J.-P. Onnela, J. Saramäki, J. Kertész, and K. Kaski, "Intensity and coherence of motifs in weighted complex networks," *Phys. Rev. E*, vol. 71, p. 065103, Jun 2005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.71.065103

[10] M. E. J. Newman, *Mathematics of Networks*. London: Palgrave Macmillan UK, 2016, pp. 1–8. [Online]. Available: https://doi.org/10.1057/978-1-349-95121-5_2565-1