

Article

Bridging the Gap between Technological Education and Job Market Requirements through Data Analytics and Decision Support Services

Evangelos Karakolis , Panagiotis Kapsalis , Stavros Skalidakis, Christos Kontzinos, Panagiotis Kokkinakos , Ourania Markaki and Dimitrios Askounis

Decision Support Systems Laboratory, School of Electrical & Computer Engineering, National Technical University of Athens, Iroon Polytechniou 9, 15780 Zografou, Greece; pkapsalis@epu.ntua.gr (P.K.); sskalidakis@epu.ntua.gr (S.S.); ckon@epu.ntua.gr (C.K.); pkokkinakos@epu.ntua.gr (P.K.); omarkaki@epu.ntua.gr (O.M.); askous@epu.ntua.gr (D.A.)

* Correspondence: vkarakolis@epu.ntua.gr



Citation: Karakolis, E.; Kapsalis, P.; Skalidakis, S.; Kontzinos, C.; Kokkinakos, P.; Markaki, O.; Askounis, D. Bridging the Gap between Technological Education and Job Market Requirements through Data Analytics and Decision Support Services. *Appl. Sci.* **2022**, *12*, 7139. <https://doi.org/10.3390/app12147139>

Academic Editors: Boris Aberšek and Andrej Flogie

Received: 23 May 2022

Accepted: 13 July 2022

Published: 15 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: In the 21st century, technology evolves extremely fast. The same applies to technology-related professions, mostly in terms of skills requirements. Contradictorily, higher education technological institutions are not always in the position to keep up with the labor market requirements. As a result, some of the skills taught in their courses are oftentimes outdated. From a learner's perspective, the main goal should be to avoid such outdated courses, as for most university students, the long-term objective is to land a job, where they will utilize the skills they acquired from their studies. On the other hand, from an educational decision maker's perspective, the most important goal is to keep up with the changes in the labor market, offering courses that will be valuable for the prospective careers of students. The work conducted in the context of this publication aims to bridge the gap between education offered in universities and job market skills' requirements in technology. Specifically, a skill and course recommender system was developed to help learners select courses that are valuable for the job market, as well as a curriculum design service, which recommends updates to a given curriculum based on the job market needs. Both services are built on top of a text mining service that retrieves job posts from several online sources and performs skill extraction from them based on text analytics techniques. Moreover, a decision support service was developed to facilitate optimal decisions for both learners and education decision makers. All services were evaluated positively by 31 early users.

Keywords: skills mismatch; higher education; technology labor market; curriculum design; data mining; recommender systems; decision support

1. Introduction and Motivation

As the labor market, especially in technology-related professions, is rapidly evolving, skills that were in demand a few years ago are now considered outdated. As a result, people who completed their studies in technology-oriented universities a long time ago and are working with technologies and tools that were highly demanded several years ago may struggle to advance their careers, or acquire a new job, despite their significant experience in the respective field in which they are employed. Therefore, an increasing number of employees are trying to advance their skillset through university or/and open courses.

On the other hand, a main challenge faced by universities is to keep up with advancements in the job market, which leads to the teaching of courses that inculcate outdated skills to students [1]. Hence, it is more important than ever for university students to select their courses in an informed and efficient manner. More specifically, they should avoid selecting courses that teach obsolete skills, and instead favor skills that are currently in high demand and will also probably be in the future.

Universities and educators themselves should also take action towards bridging the gap between the skills they teach and the labor market requirements, since their role is largely to prepare students for the job market, and failure to keep up with the advancements of technology and the labor market demand may lead to their decline. This is because more and more massive open online courses (MOOCs) are created that aim to provide state-of-the-art competencies to learners. Therefore, as it was also recognized in [2], higher education technological institutions should make significant effort to improve their curriculum based on the job market requirements.

However, it is not easy for students, learners, and university decision makers to be constantly aware of the skills that are in high demand and the ones that are considered deprecated at all given times. This is because it requires significant effort to search and draw conclusions from job market data, as there is an abundance of information in job posting websites and other resources (e.g., LinkedIn). Apart from the large volumes of data, another problem that makes such data difficult to be analyzed is that they are provided as free text, which is not easy to process. Therefore, employing text mining and analytics techniques constitutes an obvious action towards achieving a general overview of the job market.

In the context of this publication, several text analytics and data mining techniques have been utilized along with web scraping on several job posting websites in Greece, aiming to provide a general overview of the job market in terms of skill requirements. The procedure is repeated on a weekly basis to capture any changes in the job market on time. On top of this analysis, a recommender system for personalized skill and course recommendations based on job market needs has been built, as well as a tool for updating a school's curriculum. The latter identifies skills that are in high demand by the labor market but are missing from the given curriculum and proposes courses where they can be introduced, accompanied by relevant visualizations.

The Course and Skill Recommender service facilitates decisions for course selection by students and learners, making sure that the recommended courses provide valuable skills for the job market and comply with the user's interests. Moreover, even if some skills in high demand are not provided by the given curriculum, the tool recommends them so that students can acquire these skills through other sources. On the other hand, the curriculum design service facilitates decisions for updating a curriculum by recommending the introduction of missing skills to specific relevant courses, along with scores of relevance, in case there are multiple relevant courses to a skill.

Of course, both selecting a course and updating a curriculum are not only a matter of relevance and demand by the job market. Instead, such decisions depend on more criteria, such as the learner's preference for a course. Therefore, a Multi-Criteria Decision Support service was developed to facilitate multicriteria decision support capabilities for decision makers that can also be combined with the aforementioned recommendation services. More specifically, four distinct multicriteria decision support methods have been implemented that allow the user to add new criteria and obtain their importance, along with the scores obtained from the recommendation services. Thus, the user can evaluate all the alternative options through a variety of decision support methods that take into consideration all the important criteria.

This publication and the research associated with it have been conducted under the context of the EU-funded research project QualiChain [3], which aims to provide Decentralised Qualifications' Verification and Management for Learner Empowerment, Education Reengineering, and Public Sector Transformation. The results presented here refer to the Electrical and Computer Engineering (ECE) School of the National Technical University of Athens (NTUA) and the Greek software job market. However, the methodology can easily be applied to more academic institutions, regions, and scientific domains.

The current publication is structured as follows. Section 1 is the introduction to the document, presenting, in short, the current challenges in the domain and the solution that was developed to overcome them. Section 2 presents the current state of the art

by presenting other relevant publications. In Section 3, the methodology that has been followed to acquire the job market data, extract information from it, and build the Course and Skill Recommender service, the Curriculum Designer and the Multi-Criteria Decision Support service, is presented in detail. In Section 4, several examples of the aforementioned services' results are illustrated. In Section 5, an initial evaluation of the provided services by 31 early users is presented, while in Section 6, the results of the proposed services are discussed. Finally, in Section 7, the main outcomes and conclusions of this publication are presented, along with the next steps for further improving the provided services.

2. Related Work

The work that was conducted in the context of this publication has its roots in several scientific fields. Specifically, it combines text mining, web scraping, job market analysis, data mining, recommender systems, and multicriteria decision support systems to facilitate decisions in the software education domain. There are plenty of scientific publications in these fields; however, at the time of writing this publication, there were no publications building on the intersection of all the aforementioned fields. Therefore, several related publications from the most important scientific domains will be presented in brief.

Regarding job market analysis, several articles have been published, and the most pertinent ones will be presented in brief. Specifically, Pejic-Bach et al. [4] recognize the rapid advancement of the job market in Industry 4.0 (Fourth Industrial Revolution) and utilize job advertisements in order to acquire an overview of the Industry 4.0 job market, through the use of text mining. Their analysis ends up with several generic job profiles that can be divided into two categories, the ones focusing solely on Industry 4.0 and the ones focusing on management and computer science. On the other hand, Karakatsanis et al. [5] utilized text mining techniques along with the O*NET [6] database to identify the most in-demand occupations in the modern job market per industry and region. Finally, Chung et al. [7] explored the capabilities required by entry-level human resources employees utilizing text mining techniques to job posts. They found that there are four critical success factors for an entry-level human resources employee, which include specific skills, educational level, experience, and specific capabilities.

Concerning course and skill recommender systems, MOOCs are evolving rapidly in the last few years, and as a result, companies such as Coursera Inc. [8] invest significant amounts of both capital and effort to implement or improve their recommender systems. Therefore, a lot of research has been conducted in this area. For instance, Zhang et al. [9] built a recommender system for MOOCs based on distributed computation. Additionally, Jiang et al. [10] developed a novel recommendation system that suggests courses to help students prepare for target courses of interest by using a recurrent neural network recommendation approach. Finally, Elbadrawy et al. [11] investigated how student and course academic features influence the enrollment in courses and include such features to their recommendation approach with more accurate results.

Regarding skill recommendation systems, Dave et al. [12] combine skill and job recommendations. Specifically, their recommender system suggests suitable jobs, along with skills that will increase users' chances for the targeted jobs. Finally, Gughani et al. [13] built a career path recommender system, which suggests a personalized career path in terms of a graph of skills. The results pose very high precision and recall values for the obtained recommendations.

Regarding the area of curriculum design, the literature mostly includes theory towards designing a curriculum, as well as some methodologies and examples for designing specific curricula. For instance, Mary Shaw [14] presents the approach that was followed to design the curriculum of software engineering at Carnegie Mellon University. However, at the time that this publication was written, an end-to-end digital platform for the evaluation and redesign of a curriculum was missing from the literature.

In addition, there are several related publications that deal with decision support on education. For example, Vasile Gorgan [15] acknowledged the need for decision support

in higher education and performed a requirement analysis for developing data driven decision support systems for higher education. Moreover, Bresfelean et al. [16] propose a methodology for building decision support tools for academic environments to improve the quality of education processes and management.

Finally, it is worth mentioning that recommender systems are used in several cases as decision support systems since they suggest options to users according to certain criteria [17]. Therefore, the combination of recommender systems with decision support methods, such as multicriteria decision support seems to have great potential. In the context of this publication, several examples are provided to illustrate the added value of this combination.

3. Service Development Methodology

In this section, the methodology followed to build the services that were developed in the context of this publication is presented alongside an overview of the system architecture. Specifically, the services for job post data acquisition and skills extraction from job posts are presented, in conjunction with the Skill and Course Recommender service as well as the Curriculum Designer service that were built on top of the first two. Finally, the Multi-criteria Decision Support (MCDSS) service is presented along with examples of how it can be combined with the Skill and Course Recommender service and the Curriculum Designer service to facilitate optimal decision making.

3.1. System Architecture

In Figure 1, an overview of the entire system architecture is illustrated:

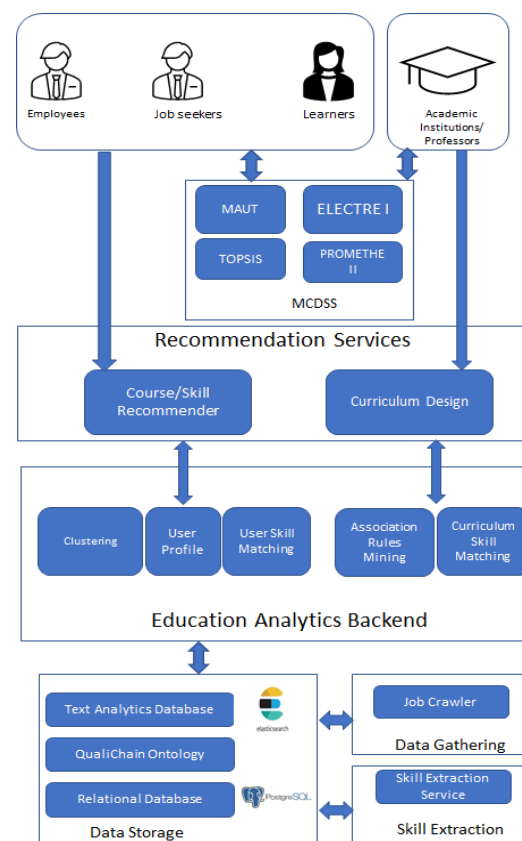


Figure 1. System architecture.

In a nutshell, JobCrawler as a data gathering service extracts data (job advertisements) from several well-known Greek job portals and stores them in a text analytics database (Elasticsearch). The results are retrieved from Elasticsearch by the Skill Extraction service,

and for each job advertisement, the latter extracts the required skills and stores the result in a relational database (PostgreSQL).

The extracted job posts and skills are retrieved from the education analytics backend in order to be used for clustering and association rules mining, to be used for optimal course and skill recommendations and curriculum update recommendations by the Course and Skill Recommender service and the Curriculum Design service. The first is used by students, learners, and employees in order to advance their careers, while the second is used by academic institutions or professors to help them update some courses provided by their academic institution or even the entire curriculum based on the needs of the job market.

Last but not least, all users can leverage the MCDSS service in order to help them make optimal decisions, taking into consideration all their criteria (even the subjective ones) through several well-known MCDSS methods. The methodology and more details about the aforementioned components are presented in the following subsections.

Specifically, in Section 3.2, JobCrawler service is presented alongside the main technologies used for data storage by the latter. In Section 3.3, the Skill Extraction service is presented. Section 3.4 presents both the backend and the final service for skill and course recommendation. Similarly, Section 3.5 presents both the backend and the frontend service for curriculum design recommendations. Lastly, Section 3.6 presents briefly the MCDSS service.

3.2. Job Posts Data Acquisition

It was noted from the initial stages of this work that to build education analytics services like the ones presented in this publication, the presence of a sufficient amount of data on the labor market was of utmost importance. Given that such data were not available in the beginning, the priority was to extract them from several sources available online. Therefore, two services were developed to acquire job posts from job posting websites.

The first service, titled JobCrawler, gets as input a job domain and a Greek job posting website and then crawls this website to gather job post data. Specifically, it performs a search for the given domain and accesses each result (job post), extracting all the relative information from it, including the description, the required skills, and the seniority level, among others. The extracted job posts are stored to an Elasticsearch engine [18] that can facilitate efficient and effective text processing and analytics. The technique of crawling web page data and extracting information out of them is called web scraping. Web scraping is performed through the Scrapy Python framework [19]. At the time of writing this publication, the supported job posting websites included Greek Indeed [20], Skywalker [21], and Kariera [22]. JobCrawler was the main tool utilized for job post acquisition in the context of this study, as the targeted education institution is located in Greece, and hence, mostly Greek job market requirements should be considered. However, it should be noted that at least in the software engineering domain, job market requirements tend to follow technological developments instead of localized geographical criteria, which means that the extracted requirements should represent software-related professions in countries other than Greece.

JobWatch is another service that was developed to extract job posts from well-known job posting websites. However, it supports only the English language and uses APIs to receive job post data. In the context of this study, it was mostly utilized to reinforce the acquired information regarding job specializations, for which the amounts of data acquired by JobCrawler were insufficient to enable an accurate analysis.

More than 1500 job posts were gathered for the software job market in Greece through the aforementioned services, to be used for further analysis.

3.3. Skill Extraction from Job Posts

After the job post acquisition phase, the job posts were available as free text. However, free text cannot facilitate decision support services without further processing. For this

reason, a Knowledge Extraction service was developed that receives as input a job post's text and extracts the skills that can be distilled from the text.

To achieve this, the first step is a text preprocessing phase. More specifically, the text is split into paragraphs and, later during the process, into tokens and bigrams. A token is an individual word, punctuation symbol or whitespace found in a text, while a bigram is a sequence of two consecutive tokens in a text. The frequency of tokens and bigrams in a text is commonly used for the statistical analysis of text.

As a next step, several part-of-speech (POS) tagging and named entity recognition (NER) techniques are applied in the identified entities as well as text matching to identify skills that are included in a list of predefined skills as listed by the ESCO classification [23]. Afterwards, duplicate entities are removed, and the result is a list of skills for each job post. The results of this service are stored in a relational database (PostgreSQL).

Of course, this is a good starting point for providing valuable decision support. However, the ESCO list of skills is very thorough, varying from highly generic skills such as programming to very rare ones, was proved to be problematic. This is because software-related jobs' skills, such as "Programming", were either explicitly declared a requirement or implied. Thus, a mechanism that would discern which skills are too generic was strongly required.

For this reason, another functionality was implemented. This classifies each identified skill as "Product", "Tool", or "Topic". As topic skills are too generic in most cases, they were not used neither by the Skill and Course Recommender service nor by the Curriculum Designer. Moreover, the frequency of each skill is counted in the entire dataset, and skills that exist in very few documents are rejected as too specific.

3.4. Skill and Course Recommender

After developing the services for job post acquisition and Knowledge Extraction, it was feasible to start building some services that can be useful to students, learners, professors, and other stakeholders in education-related issues. In this section, the Skill and Course Recommendation service is presented in brief. It was developed to facilitate personalized skill and course recommendations to students and learners. These recommendations will not only be tailored to the learner's profile. In fact, the recommended skills will also be highly desired by the job market. More specifically, the recommendation engine uses as input the resume of a learner and the information created in the Knowledge Extraction phase, classifies the resume to a few specializations of the profession that the learner wants to pursue, and proposes skills and courses that are in demand for each specialization. Finally, it recommends important skills and their respective courses from these specializations that the learner does not possess.

In order to facilitate optimal text analysis to job posts, resumes, curriculum, and course data, an Elasticsearch engine [18] was utilized. Specifically, Elasticsearch is used to facilitate advanced text search, mapping, and data aggregation to all the aforementioned text resources. These functionalities are crucial for text mining. More specifically, the available job posts were classified based on their specialization, using both advanced search techniques and manual supervision, and the resulting skills are derived from the user's closest specializations. Of course, the skills that the user already possesses are not suggested. Additionally, it is worth mentioning that the results are presented in the order of their importance to the identified specialization.

Regarding the proposed courses, they are derived from the school's curriculum. More specifically, the curriculum of a school and its courses are stored in the Elasticsearch engine, as well as the results of the Knowledge Extraction service on the courses' data. As a next step, the recommender service queries the Elasticsearch engine and retrieves the courses with the proposed skills along with a score of importance. The most relevant courses are recommended to the user, presented in the order of importance in descending order.

The aforementioned recommendation approach is not fully automated, as it requires manual classification to new job posts, which is a very time-consuming procedure. For this reason, as well as for improving the accuracy of the obtained recommendations, a

clustering recommendation approach has been implemented as well. In this approach, the identification of the different specializations is data driven, according to the required skills of the job posts. In particular, each specialization (cluster) is identified through shallow agglomerative clustering [24]. This clustering algorithm was preferred among several other clustering techniques, as it captures the different size of each cluster in contrast to other well-known clustering algorithms, such as *k*-means [25]. This means that agglomerative clustering can recognize specializations with small numbers of members as effectively as the bigger ones. Our choice is verified also from experimental results, as several clustering techniques were applied, and the most reasonable results were achieved through agglomerative clustering.

The recommended skills are derived from the three specializations (clusters) that are closer to the learner's profile. The closeness of the user to a specialization is calculated using the cosine similarity [26] between the centroid of the specific specialization and the user's profile. It is worth mentioning that both the centroid of a specialization and the profile of a learner are expressed as a vector of skills. Moreover, cosine similarity was preferred over other similarity metrics, such as the Jaccard similarity [27], as it takes into consideration both the presence and the frequency of a skill.

Finally, the selection of the recommended courses is performed in the same fashion as in the Elasticsearch approach.

The result of the clustering is very satisfactory for the software domain in Greece and the ECE School of NTUA, as illustrated in Figure 2.

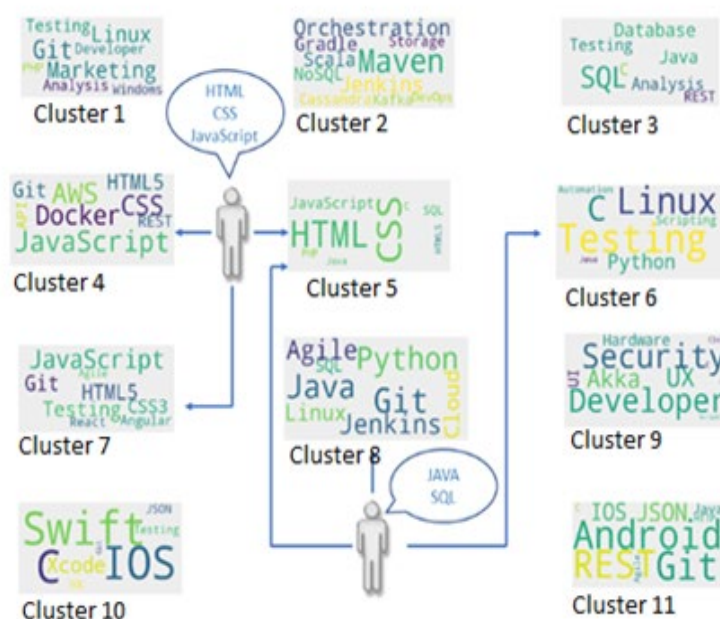


Figure 2. Clustering results [28].

The identified specializations include a cluster that could be identified as DevOps software engineer (Cluster 2), which includes skills such as Maven, Orchestration, and Gradle. Cluster 3 signifies skills such as SQL and Database with great importance, along with some programming languages with lower importance, and hence can be identified as a specialization for Database software engineer. All the clusters except Cluster 1, which shows the job posts that could not be introduced to any other specialization, are very meaningful and are close to the results of the manual classification.

More details regarding the methodology that has been followed can be found in [28].

As both approaches provide satisfactory results and due to the fact that there are cases where they provide very different recommendations, it was decided to combine them in order to provide more recommendations that may capture the entire range of the learners' preferences.

3.5. Curriculum Designer

Another service that was built on top of JobCrawler and the Skill Extraction service is the Curriculum Designer. The main objective of this service is to help higher education institutions update their curricula according to the job market requirements and, as a result, be more targeted to their students' projected careers. The Curriculum Designer takes as input a school's curriculum and the job posts of a relevant profession and identifies the skills that are highly appreciated by the job market but are not taught in the context of the given curriculum. As a next step, it identifies the courses that are suitable to introduce this skill in their syllabus. The results of this service are presented to the users through useful visualizations, along with importance scores for the skills, as well as relevance scores for the proposed courses that measure how relevant skills are to each of the courses, where they are proposed to be introduced.

The curriculum that is provided as input is analyzed and modeled as a set of courses where each course is accompanied by a list of skills that are taught in its context. In order to achieve this, the Skill Extraction service played a major role as it identified the skills taught from a course's description. Finally, the results were stored in a relational database, as well as an Elasticsearch engine for advanced text operations.

Concerning the identification of missing skills, initially, the most popular skills are identified through the Knowledge Extraction service based on the number of job posts in which they appear. Afterwards, these skills are searched in the "taught skills" section of the given curriculum. In case they cannot be found, they are considered missing, and the goal is to find courses that are relevant to these skills so that the service can recommend their introduction to these courses.

This is achieved through association rules mining [29]. With this technique, relations between skills are unraveled. More specifically, all the job posts are scanned to find the probability that two (or more) skills appear together in a job post (support metric) and to what extent finding one skill in a job post leads to a high probability of also finding another specific skill in the same job post (confidence metric). Finally, another metric is calculated, which measures to what extent the fact that two skills appear together is random (lift metric). For instance, if two skills are extremely common, it is highly probable that they will appear together, but this does not mean that they are actually related. Of course, it is very expensive in terms of computational resources to calculate these metrics for all the possible combinations of skills. Therefore, the Apriori [30] algorithm is utilized to calculate the aforementioned metrics only for sets of skills that seem related.

It is a given that the association rules cannot be calculated on each request, as it is not computationally feasible. For this reason, and since the job market demand is not changing rapidly within a short time period, the association rules are saved in a data storage to be used by the application when requested. The association rules are calculated and stored on a weekly basis. Therefore, on each request, the job market and the curriculum databases are queried to identify the missing skills, and the association rules are loaded to find relevant courses. More specifically, for each missing skill, the related skills are loaded from the association rules data, and related skills are searched in the curriculum database. Finally, if the related skills are detected in the curriculum, the courses in which they are taught are candidate courses for introducing the missing skills.

In Table 1, some examples of association rules that have been identified with high probability in Greek job posts are presented alongside the scores of support, confidence, and lift, ordered by the confidence score in descending order. The first seven are the association rules with the highest confidence score in our entire dataset. The next are very representative examples of the Greek job market. The entire list of association rules is available on the GitHub repository of the service [31].

Table 1. Association rules examples.

Antecedents	Consequents	Support	Confidence	Lift
CSS3	HTML5	0.046	0.965	10.78
HTML5, CSS	JavaScript	0.045	0.964	5.826
HTML, JavaScript	CSS	0.042	0.872	8.658
Jenkins	Git	0.051	0.857	3.635
CSS	JavaScript	0.083	0.821	4.963
Android	IOS	0.041	0.802	10.182
Docker	Git	0.062	0.797	3.381
JSON	REST	0.051	0.728	6.045
NoSQL	Python	0.041	0.598	4.945
IOS	Android	0.041	0.521	10.182
Java	SQL	0.055	0.447	2.557
Java	JavaScript	0.04	0.329	1.988

It can be observed that all these associations are what someone familiar with the software engineering sector would expect. Of course, Table 1 shows just a small subset of the identified association rules that have been identified to suggest the courses in which several missing skills should be introduced. Providing the full list of association rules would lead to a very long table that does not add significant value to this publication. Instead, more emphasis will be given in Section 4, where several examples from the results of the Curriculum Designer will be provided through the QualiChain Front End, along with meaningful visualizations, as well as the use of the Multi-Criteria Decision Support tool, to demonstrate the entire procedure up until the final decision for a user.

A more detailed description for the Curriculum Designer service methodology can be found at [32].

3.6. Multi-Criteria Decision Support Service

The Multi-Criteria Decision Support service is a general-purpose tool that facilitates the decision-making process by taking into account all the meaningful criteria. Most of the Multi-Criteria Decision Support methods can be classified into multicriteria utility methods and outranking methods that stem from the multicriteria utility theory and the outranking relations theory, respectively.

The multicriteria utility theory considers all the available criteria and calculates a total evaluation score for each alternative. The multiattribute utility theory (MAUT) is one of the multiple methods classified as a multicriteria utility method [33].

Outranking methods build a preference relation, usually called an outranking relation, among alternatives evaluated on several attributes. In most outranking methods, such as ELECTRE I and PROMETHEE II, the outranking relation is built through the execution of a series of pairwise comparisons of the alternatives [34].

As of now, the implemented Multi-Criteria Decision Support methods are the MAUT [35], TOPSIS [36], ELECTRE I [37], and PROMETHEE II [38]. To be more specific, the MAUT is a structured methodology designed to handle trade-offs among multiple objectives and is one of the oldest and most established multicriteria decision-making techniques. MAUT calculates a utility score for each alternative by calculating the sum of the normalized scores of every criterion for each alternative. The weights of each criterion and the score for each alternative for each criterion are obtained by the user. Those scores can be used to rank the alternatives.

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is based on the concept that the chosen alternative should have the shortest geometric distance from the positive ideal solution and the longest geometric distance from the negative ideal solution. TOPSIS calculates the closeness of each alternative to the ideal solution by taking into account the distances of the alternative from the ideal and negative-ideal solutions. Closeness can be used to sort the alternatives.

ELECTRE I is based on the outranking relations theory. The result of this method is a dominance matrix in which the element (i, j) indicates whether or not alternative i outranks alternative j . The result of ELECTRE I can be represented by a graph in which the alternatives are represented by nodes, while the existence of an edge from node i to node j denotes that alternative i dominates alternative j . This method cannot be used to rank the alternatives but is used to show relative dominance between alternatives.

PROMETHEE II is also based on the outranking relations theory. It considers the net outranking flow, which is the balance between the positive and the negative outranking flows. This method provides complete ranking of the alternatives. In comparison with ELECTRE I, it extends the notion of a criterion in terms of how strict it is. Hence, a criterion can be strict, linear, or strict with an indifference region, meaning that the criterion becomes strict if it takes values above the aforementioned threshold. The resulting outranking relations are less sensitive to slight differences in the scores of the criteria, and as a result, it is easier to interpret. Last, it can be used for ranking of the alternatives, which is much appreciated by its users.

In the following section, the Multi-Criteria Decision Support service results are presented on top of the results of the Skill and Course Recommender service to facilitate the decision of selecting courses according to the labor market requirements and the learner's profile. Moreover, the scenario of introducing a new skill to a curriculum by using the suggestions of the Curriculum Designer, along with the results of the Multi-Criteria Decision Support service, is presented to facilitate the decision of introducing a new skill to the curriculum and the selection of the most suitable courses for the new skill to be introduced.

4. Service Demonstration

In this section, both the Skill and Course Recommender service and the Curriculum Designer alongside the Multi-Criteria Decision Support service are presented through actual examples. Specifically, the decision for the optimal selection and prioritization of courses for an ECE student is facilitated through the Course Recommender and the Multi-Criteria Decision Support service. In addition, the decision for updating the ECE School's curriculum based on the labor market needs is facilitated by the Curriculum Designer service and the Multi-Criteria Decision Support service.

4.1. Course Selection

As stated in the previous section, in order for the service to provide course and skill recommendations to a user, the latter should upload a detailed CV, which includes the target sector, the acquired skills, and the work experience, to the service. In this specific example, let us assume that a user named Mark targets the information technology sector, and he is experienced in Java and MySQL, as illustrated in Figure 3 (CV upload). As observed, his proficiency level in Java is medium, as he has 2 years of experience with Java, and advanced for MySQL with 3 years of experience. It is worth mentioning that the * symbol in the form of Figure 3 indicates fields that are mandatory. This is the case also for other forms that are presented.

After saving the uploaded CV, the service calculates the most appropriate skills for Mark based on his profile. The recommended skills for Mark are illustrated in Figure 4:

As presented in Figure 4, the recommended skills that Mark does not seem to possess are Git, Docker, Jira, SQL, Databases, Azure, Cloud, and Jenkins. There are more recommended skills for the specific user, but only the most important, in terms of preference by the labor market, are presented in descending order according to the demand for each one of them.

Skill 1 Delete

Label *
Java

Proficiency level *
Medium

Comment *
2 years of experience

Skill 2 Delete

Label *
MySQL

Proficiency level *
Advanced

Comment *
3 years of experience

Save Close

Figure 3. CV uploaded by learner.

Filter

Skill	Action
git	▲ View
docker	▲ View
jira	▲ View
sql	▲ View
database	▲ View
azure	▲ View
cloud	▲ View
jenkins	▲ View

Figure 4. Recommended skills.

As a next step, Mark can either decide which skills he should try to learn through the help of the Multi-Criteria Decision Support service or receive course recommendations from his university. As the course selection is tightly coupled with the targeted skills, the user chooses to receive course recommendations as presented in Figure 5:

Filter

Name	Description	Score	Action
Databases	Database Management Systems and their architecture. Data Structures for Databases. Modeling - The E-R model. Reference to classic Database Models (Hierarchical, Network). The Relational Model. Database Languages - The SQL language. File Systems and Physical Database Design. Logical Design and Normalization. Management and Operation Issues (Integrity, optimization, reorganization, security, functionality, etc.). Other Issues (Object Oriented Systems, Multi-Systems, Personal Computer Systems, etc.).	1	View
Operating Systems	This course covers the general principles of Operating Systems. Evolution of Operating Systems. Input/Output, concurrent processes, critical section, process synchronization and communication. CPU scheduling. Memory management (static and dynamic allocation, virtual memory, paging, segmentation). File management, disk scheduling, deadlocks.	1	View
Distributed Systems	Synchronization: Natural, logical and vector clocks, natural and logical clock synchronization, Lamport's algorithm. Distributed Coordination: Mutual Exclusion Algorithms, Election Algorithms, Distributed Agreement, Interactive Consistency, Byzantine Protocols, Distributed Consensus Algorithm Paxos. Tasks and concurrency control: ACID properties, embedded transactions, strict two-phase locking, concurrent problems, timer layout, optimistic concurrency control. Distributed Transactions and Timing Control. Distributed Individual Transactions. Distributed Timing Control. Deadlock Detection. Error Recovery. Data Backup: Error Tolerance. Primary / Secondary Backup, Passive / Active Backup. Network Partition Management. High Availability, Consistency Levels, CAP Theorem. Peer-to-peer networks, distributed hash tables, distributed storage systems, Map Reduce and Bulk Synchronous Parallel distributed processing models.	1	View
Operating Systems Laboratory	This course offers laboratory projects to assimilate critical concepts from the previous courses "Computer Architecture" and "Operating Systems". The course includes 3-4 exercises in distributed UNIX programming, development of device drivers, kernel-level programming, and extensions of the Linux OS.	1	View
Human-Computer Interaction	The course deals with the design, development and evaluation methodologies of computer systems that interact with users to a significant degree. The purpose is to review theoretical models, technologies, methods and tools for the design and development of interactive computing systems. The course includes (1) Human modeling of computer system and conceptual interaction framework, (2) Cognitive models, perception and representation, attention and memory representation and organization of knowledge(3) Interactive Systems Design Methodologies: Human-centric Design, Usability Requirements, Task Analysis, GOMS Models, Dialogue Methods, Interface Design, Usability and Accessibility of Internet Services, Interactive Systems Evaluation Techniques, (4) Overview of Interactive Technologies and Applications: Distributed Intelligence Systems, Ubiquitous Computer, Augmented Reality Systems, Synthetic Animation, Accessibility Interface Systems. Collaboration Applications.	0.87	View

Figure 5. Recommended courses.

As observed in Figure 5, the recommended courses are Databases, Operating Systems, Distributed Systems, Operating Systems Laboratory, and Human–Computer Interaction. Additionally, Multimedia Technology and Computer Systems Performance are among the results but not visible in Figure 5. The resulting courses are ordered by their relevance score, which is also illustrated in Figure 5.

Mark is more interested in the courses Databases, Operating Systems, Distributed Systems, and Human–Computer Interaction. Moreover, he wants to select courses not only based on the relevance score but also based on the difficulty of the course, his personal interests, and to what extent the skills he will acquire from the selected course are considered state of the art. Therefore, in order to decide which course to select, he uses the Multi-Criteria Decision Support service, as illustrated in Figure 6.

+ Add criteria + Add alternative ➤ Send data

Collect a method

MAUT

TOPSIS

ELECTRE I

PROMETHEE II

	Criteria name *	Criteria name *	Criteria name *	Criteria name *
	Relevance	Difficulty	State of the Art	Interest
	Weight *	Weight *	Weight *	Weight *
	40	15	20	25
	Type *	Type *	Type *	Type *
	0	1	0	0
	Alternative name *	Value *	Value *	Value *
<input checked="" type="checkbox"/>	Databases	100	50	50
<input checked="" type="checkbox"/>	Operating Systems	100	40	50
<input checked="" type="checkbox"/>	Distributed Systems	100	80	80
<input checked="" type="checkbox"/>	Human Computer Interaction	87	30	40

Figure 6. Multi-Criteria Decision Support service input.

The user selects the MAUT method, and the weights of the criteria are 40% for relevance, 15% for difficulty, 20% for state of the art, and 25% for interest. The difficulty criterion should be minimized, while all the other criteria should be maximized with the selected course. The relevance scores are the scores obtained by the Course Recommender, while the other scores are obtained by the user according to his perception for them. The result is illustrated in Figure 7.

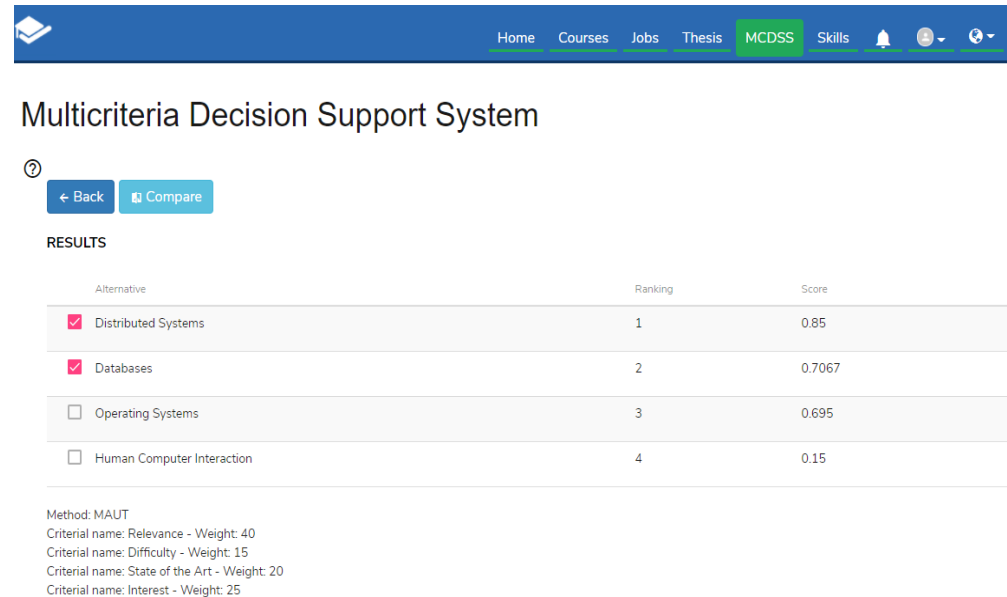


Figure 7. MAUT method results.

As shown in Figure 7, the most suitable courses for the user are Distributed Systems and Databases, while the Operating Systems course follows with a slightly smaller overall score, and Human–Computer Interaction is the last choice with a much smaller overall score. Of course, the user might need to dig further to decide which course to choose. By selecting two courses, he can further receive bipartite comparisons, comparing them for every criterion, as shown in Figure 8 for the courses Databases and Distributed Systems.

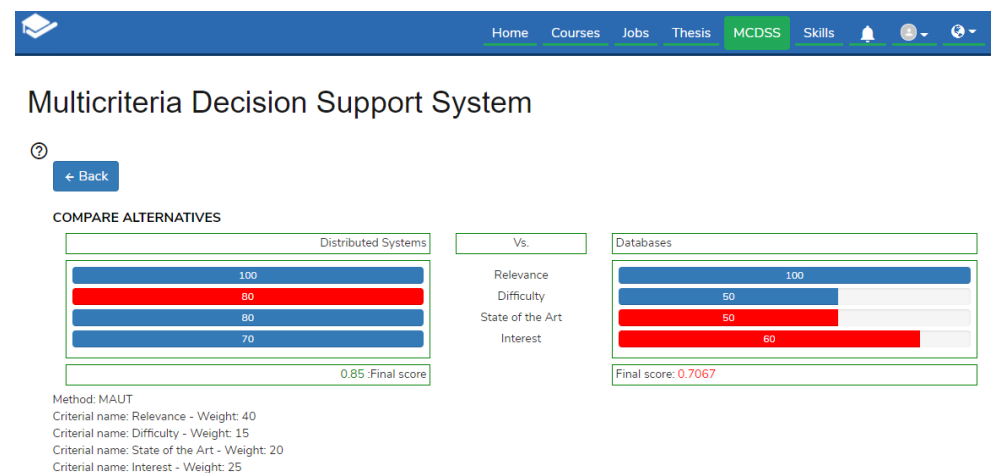


Figure 8. Bipartite comparison between courses.

As observed, the Databases course is considered to be easier, but it is less interesting and also less effective in providing state-of-the-art skills. Thus, the Distributed Systems course seems to be the best choice.

The procedure is similar for users who want to select skills on which they should focus, and therefore, it is not presented in the context of this publication.

4.2. Updating a Curriculum

In this section, the procedure for curriculum updates based on the requirements of the labor market is presented. Specifically, a professor, in collaboration with the board of the ECE School, intends to examine the skills that are in high demand by the labor market but are not taught in the current curriculum. In the figure below (Figure 9), it can be observed that about 60% of the most desirable skills by the labor market are covered by the curriculum, while in the upper-left section of the page, the missing skills, along with recommendations of courses where they can be introduced, are presented.

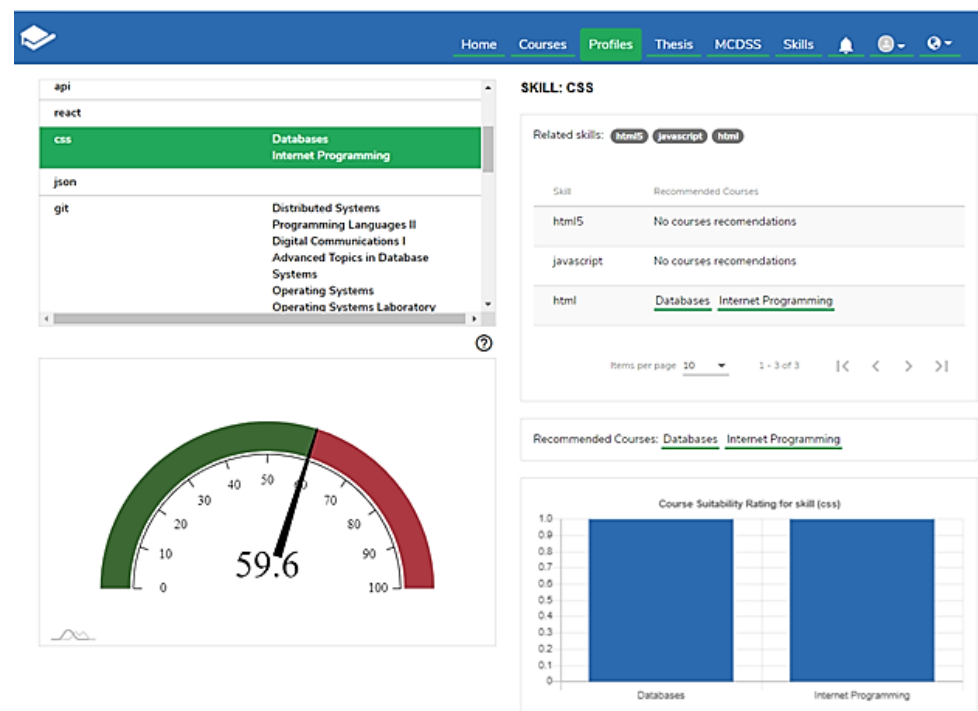


Figure 9. Curriculum Designer analytics page, emphasizing CSS skill.

More specifically, APIs, React, and JSON are not considered relevant for introduction in any course of the curriculum, while CSS seems to be relevant with the courses Databases and Internet Programming, with the same relevance score, since both are related to the skill HTML, as illustrated in the right section of the web page. On the other hand, Git is suggested for introduction in several courses, including Distributed Systems, Programming Languages II, Digital Communications, and Advanced Topics in Database Systems. In Figure 10, the input for the Multi-Criteria Decision Support service is illustrated. The decision to be made is the introduction of CSS in either the course of Databases or the one of Internet Programming. The selected method is PROMETHEE II, while the criteria are the relevance between CSS and each course, which is the most important (45% importance); the workload of each course (30% importance); and the Convenience (25%) to change each of the examined courses. As observed, Internet Programming is much more relevant to the CSS skill than Databases, but the latter's workload is smaller than the former's. Additionally, the Databases course is more convenient to be updated. It is worth mentioning that the preference value shows the intensity of the preference for one alternative over another, while indifference is a threshold, which decides the size of the difference so that an alternative can be considered superior to another.

Select a method
PROMETHEE II

	Criterion name *	Weight *	Type *	Preference *	Indifference *	Criteria *
	Relevance	45	0	20	10	usual
	Workload	30	1	25	0	usual
	Convenience	25	0	10	20	usual
Alternative name *	Value *					
Internet Programming	100	80				
Alternative name *	Value *					
Databases	50	70				

Figure 10. Multi-Criteria Decision Support input for skill introduction in a course.

As illustrated in Figure 11, the Databases course seems to be a slightly better alternative, even if the course is inferior in terms of relevance. Of course, the result will be different if the preference and indifference thresholds are modified. The same is true for the selected method.

Multicriteria Decision Support System

?

← Back Compare

RESULTS

Alternative	Ranking	Score
✓ Databases	1	0.10000000000000003
✓ Internet Programming	2	-0.10000000000000003

Method: PROMETHEE II
 Criterion name: Relevance - Weight: 45
 Criterion name: Workload - Weight: 30
 Criterion name: Convenience - Weight: 25

Figure 11. Results of PROMETHEE II for introducing CSS in the Databases or Internet Programming course.

For instance, in the following figure (Figure 12), the TOPSIS method is applied for the same alternatives and criteria, and the best option (Figure 13) seems to be Internet Programming.

+ Add criteria
+ Add alternative
> Send data

Select a method
 TOPSIS

	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #f08080; width: 20px; height: 20px; margin-bottom: 5px;"></div> <div>Criterial name *</div> <div>Relevance</div> <hr/> <div>Weight *</div> <div>45</div> <hr/> <div>Type *</div> <div>0</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #f08080; width: 20px; height: 20px; margin-bottom: 5px;"></div> <div>Criterial name *</div> <div>Workload</div> <hr/> <div>Weight *</div> <div>30</div> <hr/> <div>Type *</div> <div>1</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #f08080; width: 20px; height: 20px; margin-bottom: 5px;"></div> <div>Criterial name *</div> <div>Convenience</div> <hr/> <div>Weight *</div> <div>25</div> <hr/> <div>Type *</div> <div>0</div> <hr/> </div>		
<div style="background-color: #f08080; width: 20px; height: 20px; margin: 0 auto;"></div>	<div style="border: 1px solid black; padding: 5px;"> <div>Alternative name *</div> <div>Internet Programming</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div>Value *</div> <div>100</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div>Value *</div> <div>80</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div>Value *</div> <div>50</div> <hr/> </div>	
<div style="background-color: #f08080; width: 20px; height: 20px; margin: 0 auto;"></div>	<div style="border: 1px solid black; padding: 5px;"> <div>Alternative name *</div> <div>Databases</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div>Value *</div> <div>50</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div>Value *</div> <div>70</div> <hr/> </div>	<div style="border: 1px solid black; padding: 5px;"> <div>Value *</div> <div>60</div> <hr/> </div>	

Figure 12. TOPSIS input.

RESULTS

Alternative	Ranking	Score
<input type="checkbox"/> Internet Programming	1	0.8251
<input type="checkbox"/> Databases	2	0.1749

Method: TOPSIS
 Criterial name: Relevance - Weight: 45
 Criterial name: Workload - Weight: 30
 Criterial name: Convenience - Weight: 25

Figure 13. TOPSIS results.

After scrolling down to the missing skills section, as illustrated in Figure 14, IOS, JavaScript, and Akka are also missing from the curriculum, while they are highly demanded by the job market. For IOS, no relevant course can be found in the curriculum, while for Akka, the only relevant course is Human–Computer Interaction. On the other hand, JavaScript is associated with several skills (both taught and missing from the curriculum), and as a result, it can be introduced in several courses. Specifically, it is associated with the skills HTML, Git, HTML5, SQL, CSS, Java, and REST, and is compatible with the courses Databases and Internet Programming with higher probability, as well as with the courses Multimedia Technology, Computer Systems Performance, Distributed Systems, Programming Languages I and II, Software Engineering, and Compilers, with lower probability. The latter courses along with Internet Programming, in the context of their syllabi, are teaching Java, which is highly associated with JavaScript, while Internet Programming and Databases are also teaching HTML, which is highly associated with JavaScript as well. Finally, JavaScript is also highly associated with SQL, which is taught in the course Databases.

The process of inserting the JavaScript skill in a course is the same as the process that was already presented.

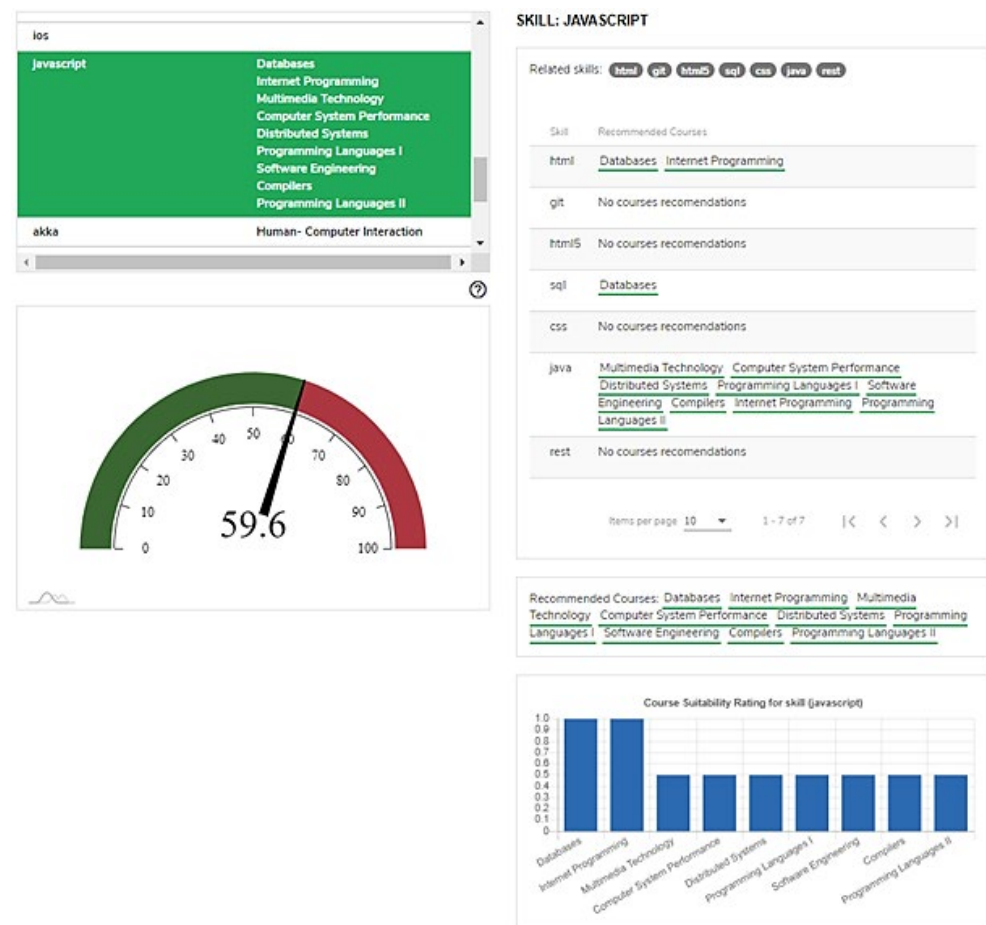


Figure 14. Curriculum Designer analytics page, emphasizing the JavaScript skill.

5. Evaluation by Early Users

It is worth mentioning that all the presented services have been evaluated by an initial set of early users that were employed to use and evaluate them. Specifically, 28 university students of the ECE School of NTUA were requested to use the Course Recommender service to help them select courses, while 3 professors were requested to use the Curriculum Designer service to receive recommendations concerning updates for their courses. Finally, all 31 users were requested to use the MCDSS service to prioritize their alternatives. For all three services, the majority of users found them at least useful, according to the questionnaires that they were requested to fill in after using the services.

In particular, the vast majority of Course Recommender evaluators stated that the recommended skills are in high demand by the labor market, and the proposed courses teach the recommended skills. Moreover, the vast majority of the evaluators stated that the recommended skills and courses are at least satisfactory to them. Last but not least, they stated that it is very likely to use Course Recommender in the future for course selection. These results are illustrated in more detail also in Figure 15:

Regarding Curriculum Designer, all three evaluators stated that the proposed skills are highly demanded by the labor market, and that they are relevant to the courses in which they are proposed to be introduced. Finally, all professors stated that it is very likely to use Curriculum Designer for updating their courses in the future. The evaluation results for Curriculum Designer are illustrated in Figure 16:

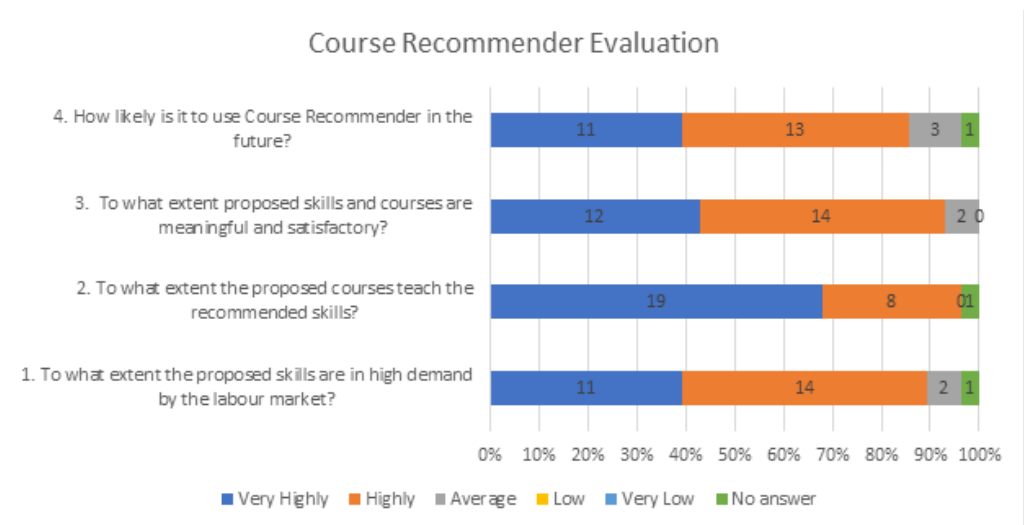


Figure 15. Course Recommender evaluation by university students.

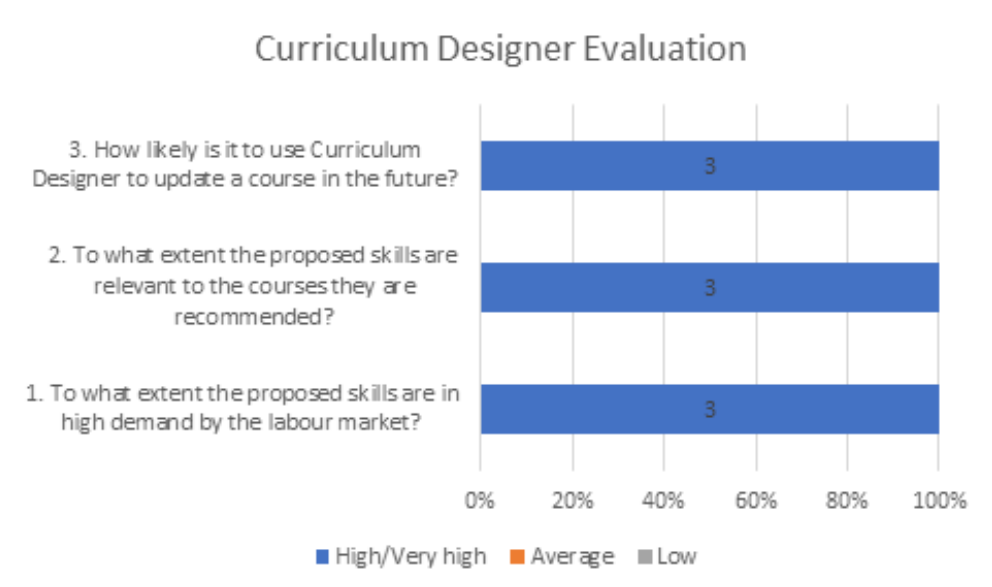


Figure 16. Curriculum Designer evaluation by professors.

Finally, the MCDSS service was found to be useful for course selection by most of the students. The same goes for professors. Specifically, 16 out of 28 students and 2 out of 3 professors found the MCDSS service useful for updating a course, while 10 students and 1 professor were uncertain about its usefulness, and 2 students found it useless. Finally, regarding the likeliness of using the MCDSS service in the future, almost half of the users were positive. However, there were a considerable number of users (1 out of 3) who were not sure if they would use it. This can partly be explained by the fact that MCDSS methods require users to be somewhat familiar with them so that they can be effectively used. Finally, there were 3 users who do not intend to use the service in the near future. The evaluation results for the MCDSS service are illustrated in Figure 17:

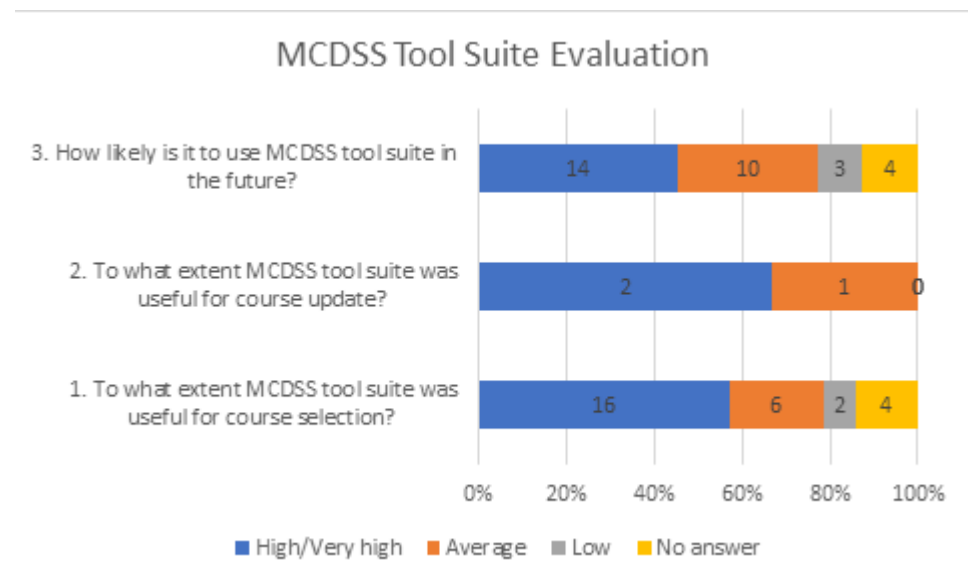


Figure 17. MCDSS service evaluation.

6. Discussion

As illustrated in Section 4, both skill and course recommendations as well as recommended curriculum updates are very meaningful for users. Specifically, course and skill recommendations help learners ensure that the skills they are being taught are both valuable for the labor market and compatible with their current profile. Without using this service, they would spend several days to draw valuable inferences regarding the labor market requirements, while their analysis would not be built on a sufficiently large data sample, and therefore, it might not be as accurate as the results of the Skill and Course Recommendation service. Of course, in case users need more information about specific specializations in the job market, they can conduct their own analyses, since the objective of the Skill and Course Recommendation service is to provide a generic overview of the labor market and be used as an advisor. These objectives are greatly achieved, according to the opinions of the early users of the service.

Concerning comparison with other course recommendation services from the literature, the results of this study are not directly comparable. This is because this study is based on real-life data from the Greek labor market, and at the time of authoring this publication, only a limited number of users were using the platform, and their reactions to the recommendations were not tracked. For instance, in [39], Gulzar et al. developed a personalized course recommender system based on a hybrid approach, which performs well in comparison with other similar services. The performance value they use is accuracy of recommendations as the ratio between the number of relevant courses proposed and the total number of courses proposed. For this publication, information on the relevance of a specific recommendation to a user's preferences is not available. However, according to the questionnaire in Figure 15, the recommended courses are very relevant with the users' preferences, as well as compatible with the requirements of the job market.

On the other hand, Al-Badarnah et al. [40] evaluated their course recommender system in terms of precision (the ratio between the number of recommended course taken by students and the number of recommended courses) and recall (the ratio between the number of recommended course taken by students and the number of courses taken by students), with satisfactory results. However, such information requires either a big user base, or monitoring users for a long period, in order to draw significant conclusions about the courses taken by the students. For similar reasons the results of our study are not comparable with the results in [41]. Specifically, the authors evaluate their system in terms of hit ratio (HR) and normalized discounted cumulative gain (NDCG). Similarly, due

to lack of information of specific user reactions to the provided recommendations, we cannot calculate these evaluation metrics. Hence, the evaluation of the presented services using precision, recall, NDCG, and HR is intended to be performed in future extensions of this work.

Regarding the Curriculum Designer service, it helps educational institutions be aware in a few seconds whether there are skills in high demand by the job market that are not taught in the context of their curriculum. Moreover, educational institutions that use the Curriculum Designer are aware of new skills that are compatible with a course and can be introduced in its syllabus. The use of this service can also save a lot of time for education institutions' decision makers, as updating a school's curriculum is a crucial decision and, therefore, much more time-consuming than selecting a course. Furthermore, it helps higher education advance in parallel with the labor market. This is very important, since the labor market, especially in technology-related fields, is evolving more rapidly than ever, while educational institutions tend to advance at a very slow pace. As such, this service can help them monitor and quantify the advancement of the labor market, making it easier for the educational institutions to follow. An initial set of early users agrees with these conclusions, as illustrated also in the previous section.

As mentioned also in Section 2, there is a lack of end-to-end curriculum design services in order to compare the results of the presented study. Furthermore, other publications related to curriculum design services focus on the methodology, while the results are rarely evaluated. For instance, in [42], the authors present a methodology for designing a data science curriculum leveraging an ontology for data science competences and present an example of the workflow for the design of a curriculum alongside some results. However, the evaluation of their results is missing.

Last but not least, for both services, the users can accompany their provided recommendations with decision support methods that can help them make their final decision in an informed fashion that considers all the subjective criteria that are meaningful to them. Moreover, they can use several alternative methods that stem from different theories and end up with the one that is the most suitable for them.

As a last note, it is worth mentioning that although this approach was applied only for the Greek job market and the curriculum of the ECE School of NTUA, it can be applied to more schools and regions with little to no modifications for the software job market or with slight modifications for other domains.

7. Conclusions and Future Work

In this paper, the gap between higher education and job market requirements for the software industry in Greece was identified. In this context, text mining and web scraping techniques were applied to both job posting websites in Greece and the curriculum of the ECE School of NTUA, in order to monitor this gap in terms of skills taught by the school and skills required by the software job market in Greece, by gathering job posts and courses. As a result, more than 1500 job posts and more than 100 courses were gathered, while several required (from job posts) and taught (from courses) skills were extracted.

On top of these results, a course and skill recommender system was developed to provide students and learners with personalized course and skill recommendations that will help them enhance their position in the software job market, as on the one hand, the skills that are recommended are highly appreciated by the latter, and on the other hand, the recommendations are tailored to their profile and the specializations for which they seem to be more suitable.

Furthermore, a curriculum design service was developed. It recommends updates on the given curriculum to make it more compatible with the job market requirements. More specifically, it identifies the most in-demand skills from the labor market that are missing from the curriculum and suggests their introduction in compatible courses, in case such courses exist, along with compatibility scores.

On top of both of the aforementioned services, Multi-Criteria Decision Support can be applied through the use of the Multi-Criteria Decision Support service, which was developed to facilitate better decision making for users and enable them to introduce more criteria than the ones obtained by the aforementioned recommendation services.

The Course and Skill Recommender service appears to provide valuable recommendations. It helps learners not only make informed decisions, but also save time spent on doing research for the courses that they should select. Similar benefits are provided to educational institutions' stakeholders by the Curriculum Designer service. Specifically, they can be aware at any time of the skills that are in high demand by the software job market, make informed decisions on curriculum updates, and save time that would be spent doing research on the job market. Finally, both students and educational stakeholders can take advantage of the Multi-Criteria Decision Support service to reinforce their decisions.

Feedback by the evaluators was positive for all three services in terms of both accuracy and usefulness and satisfaction. Additionally, the majority of the evaluators stated that they intend to use the presented services in the future.

Concerning the future, the implementation team will focus on improving the obtained course and skills recommendations and the proposed curriculum updates by trying new recommendation and data mining approaches and technologies, such as collaborative filtering.

Furthermore, as user reactions to recommendations will be made available in the near future, and more users plan to use the presented services, different metrics will be utilized for the evaluation of the Course and Skill Recommendation service, including precision, recall, and NDCG.

Additionally, since the number of early users of the system is limited and further evaluation is required in order to have reliable results, once a significant number of users adopt the provided services, further evaluations through surveys will be performed by the users.

Another possible extension of this work for the future has to do with monitoring the users' careers over time. Specifically, it is intended to monitor several users' career evolution over time in order to decide whether the course recommendations they received actually helped them with their careers. Additionally, the latter's career evolution will be compared with the evolution of students who did not use the recommendation services described on this paper to evaluate to what extent the proposed Skill and Course Recommendation service is beneficial to the students.

Moreover, new methods will be examined to be introduced to the Multi-Criteria Decision Support service based on the suggestions of additional users of the service. Finally, the presented methodology and services will be applied and evaluated for several new domains and universities apart from the software engineering labor market and the ECE School of NTUA.

Furthermore, with the introduction of more schools and domains in the system, a comparison between different schools and different domains will be feasible, which can be very useful for facilitating the selection between different schools or domains by learners. Finally, as more courses from different universities will be added to the Course Recommender service, fairness among course providers in terms of their items' exposure to users will be examined following methodologies such as the one proposed in [43].

Author Contributions: Conceptualization, E.K.; methodology, E.K. and P.K.; software, E.K., P.K. and S.S.; validation, P.K. (Panagiotis Kokkinakos), O.M. and D.A.; formal analysis, E.K. and C.K.; investigation, E.K. and P.K.; data curation, P.K. and S.S.; writing—original draft preparation, E.K.; writing—review and editing, E.K., C.K., O.M. and P.K. (Panagiotis Kokkinakos); visualization, P.K. (Panagiotis Kapsalis) and S.S.; supervision, D.A.; project administration, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was cofunded by the European Union's Horizon 2020 research and innovation program under QualiChain, Grant Agreement No. 822404.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets that were used for this publication include job advertisements that were gathered through QualiChain/JobCrawler, which is available in GitHub [44]; the curriculum of the Electrical and Computer Engineering School of the National Technical University of Athens, which is publicly available [45]; and the CVs of several individuals, which are not publicly available due to data privacy restrictions. It is worth mentioning that the procedure for gathering individuals' CVs was conducted following the provisions of GDPR.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Kontzinos, C.; Markaki, O.; Kokkinakos, P.; Karakolis, V.; Skolidakis, S.; Psarras, J. University process optimisation through smart curriculum design and blockchain-based student accreditation. In Proceedings of the 18th International Conference on WWW/Internet, Cagliari, Italy, 7–9 November 2019. [CrossRef]
- Kontzinos, C.; Karakolis, V.; Skolidakis, S.; Markaki, O.; Androutsopoulou, M.; Psarras, J. Combining Blockchain, semantics and data analytics for university process optimization. *IADIS Int. J. Comput. Sci. Inf. Syst.* **2019**, *2*, 59–77.
- QualiChain Consortium. QualiChain Project. 2019. Available online: <https://qualichain-project.eu> (accessed on 10 May 2022).
- Pejic-Bach, M.; Bertonecel, T.; Meško, M.; Krstić, Ž. Text mining of industry 4.0 job advertisements. *Int. J. Inf. Manag.* **2020**, *50*, 416–431. [CrossRef]
- Karakatsanis, I.; AlKhader, W.; MacCrory, F.; Alibasic, A.; Omar, M.A.; Aung, Z.; Woon, W.L. Data mining approach to monitoring the requirements of the job market: A case study. *Inf. Syst.* **2017**, *65*, 1–6. [CrossRef]
- National Research Council. *A Database for a Changing Economy: Review of the Occupational Information Network (O*NET)*; National Academies Press: Washington, DC, USA, 2010.
- Chung, C.-H.; Chen, L.-J. Text mining for human resources competencies: Taiwan example. *Eur. J. Train. Dev.* **2019**, *45*, 588–602. [CrossRef]
- Coursera. Coursera—Learn without Limits. Available online: <https://about.coursera.org/> (accessed on 10 May 2022).
- Zhang, H.; Huang, T.; Lv, Z.; Liu, S.; Zhou, Z. MCRS: A course recommendation system for MOOCs. *Multimed. Tools Appl.* **2018**, *77*, 7051–7069. [CrossRef]
- Jiang, W.; Pardos, Z.A.W.Q. Goal-based course recommendation. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, Tempe, AZ, USA, 4–8 March 2019. [CrossRef]
- Elbadrawy, A.; Karypis, G. Domain-aware grade prediction and top-n course recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 183–190. [CrossRef]
- Dave, V.S.; Zhang, B.; Al Hasan, M.; AlJadda, K.; Korayem, M. A combined representation learning approach for better job and skill recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge management, Torino, Italy, 22–26 October 2018. [CrossRef]
- Gugnani, A.; Reddy Kasireddy, V.K.; Ponnalagu, K. Generating unified candidate skill graph for career path recommendation. In Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018. [CrossRef]
- Shaw, M. *Software Engineering for the 21st Century: A Basis for Rethinking the Curriculum*; Carnegie Mellon University: Pittsburgh, PA, USA, 2005.
- Gorgan, V. Requirement analysis for a higher education decision support system. Evidence from a Romanian University. *Procedia-Soc. Behav. Sci.* **2015**, *197*, 450–455. [CrossRef]
- Bresfelean, V.P.; Ghisoiu, N.; Lacurezeanu, R.; Sitar-Taut, D.-A. Towards the development of decision support in academic environments. In Proceedings of the ITI 31st International Conference on Information Technology Interfaces, Dubrovnik, Croatia, 22–25 June 2009. [CrossRef]
- Nunes, I.; Jannach, D. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Model. User-Adapt. Interact.* **2017**, *27*, 393–444. [CrossRef]
- ElasticSearch. Elasticsearch: The Official Distributed Search & Analytics Engine. Available online: <https://www.elastic.co/elasticsearch/> (accessed on 10 May 2022).
- Scrapy Framework. Scrapy | A Fast and Powerful Scraping and Web Crawling Framework. Available online: <https://scrapy.org/> (accessed on 10 May 2022).
- Indeed Job Portal in Greece. Indeed. Available online: <https://gr.indeed.com/> (accessed on 10 May 2022).
- Skywalker Job Portal. Skywalker. Available online: <https://www.skywalker.gr> (accessed on 10 May 2022).
- Kariera Job Portal. Kariera Hiring Solutions. Available online: <https://www.kariera.gr> (accessed on 10 May 2022).
- European Commission. What Is ESCO? Available online: <https://ec.europa.eu/esco/portal/home> (accessed on 10 May 2022).
- Müllner, D. Modern hierarchical, agglomerative clustering algorithms. *arXiv* **2011**, arXiv:1109.2378.
- Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [CrossRef]

26. van Dongen, S.; Enright, A.J. Metric distances derived from cosine similarity and Pearson and Spearman correlations. *arXiv* **2012**, arXiv:1208.3145.
27. Zhang, X.-L.; Fu, Y.-Z.; Chu, P.-X. Application of Jaccard Similarity coefficient in recommender system. *Comput. Technol. Dev.* **2015**, *24*, 158–165.
28. Karakolis, E.; Kapsalis, P.; Skolidakis, S.; Kontzinos, C.; Kokkinakos, P.; Markaki, O.; Vlachou, S.; Psarras, J. QualiChain Recommender—Providing personalised skill and course recommendations based on labour market needs. In Proceedings of the 13th Annual International Conference of Education, Research and Innovation, Online, 9–10 November 2020. [[CrossRef](#)]
29. Zhao, Q.; Sourav, S.B. *Association Rule Mining: A Survey*; Nanyang Technological University: Singapore, 2003.
30. Borgelt, C.; Kruse, R. Induction of association rules: Apriori implementation. In *Compstat 2002, Proceedings of the 15th Symposium in Computational Statistics, Berlin, Germany, 24–28 August 2002*; Springer: Cham, Switzerland, 2002; pp. 395–400. [[CrossRef](#)]
31. QualiChain Curriculum Designer Github Repository. Available online: https://github.com/QualiChain/curriculum_designer_ (accessed on 10 May 2022).
32. Karakolis, E.; Kapsalis, P.; Skolidakis, S.; Kontzinos, C.; Kokkinakos, P.; Markaki, O.; Vlachou, S.; Psarras, J. QualiChain Curriculum Designer—Providing Intelligent Recommendations for Curriculum Restructuring based on Labour Market Needs. In Proceedings of the 13th Annual International Conference of Education, Research and Innovation, Online, 9–10 November 2020. [[CrossRef](#)]
33. Dyer, J.; Fishburn, P.C.; Steuer, R.E.; Wallenius, J.; Zionts, S. Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years. *Manag. Sci.* **1992**, *38*, 645–654. [[CrossRef](#)]
34. Bouyssou, D.; Philippe, V. Ranking Alternatives on the Basis of Preference Relations: A Progress Report with Special Emphasis on Outranking Relations. *J. Multi-Criteria Decis. Anal.* **1997**, *6*, 77–85. [[CrossRef](#)]
35. Dyer, J.S. Maut—Multiattribute Utility Theory. In *Multiple Criteria Decision Analysis: State of the Art Surveys*; Springer: New York, NY, USA, 2005; pp. 265–292. [[CrossRef](#)]
36. Çelikkbilek, Y.; Tüysüz, F. An in-depth review of theory of the TOPSIS method: An experimental analysis. *J. Manag. Anal.* **2020**, *7*, 281–300. [[CrossRef](#)]
37. Figueira, J.; Greco, S.; Ehrogott, M. Electre Methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*; Springer: Boston, MA, USA, 2005; pp. 133–153.
38. Brans, J.P.; Mareschal, B. The Promethee Methods for MCDM. In *Readings in Multiple Criteria Decision Aid*; Springer: New York, NY, USA, 1990; pp. 216–252.
39. Gulzar, Z.; Leema, A.; Deepak, G. PCRS: Personalized Course Recommender System Based on Hybrid Approach. *Procedia Comput. Sci.* **2018**, *125*, 518–524. [[CrossRef](#)]
40. Al-Badarenah, A.; Alsakran, J. An automated recommender system for course selection. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *7*, 166–175. [[CrossRef](#)]
41. Li, Q.; Kim, J. A Deep Learning-Based Course Recommender System for Sustainable Development in Education. *Appl. Sci.* **2021**, *11*, 8993. [[CrossRef](#)]
42. Demchenko, Y.; Communiello, L.; Reali, G. Designing customisable data science curriculum using ontology for data science competences and body of knowledge. In Proceedings of the International Conference on Big Data and Education, London UK, 30 March–1 April 2019; pp. 124–128. [[CrossRef](#)]
43. Karakolis, E.; Kokkinakos, P.; Askounis, D. Provider Fairness for Diversity and Coverage in Multi-Stakeholder Recommender Systems. *Appl. Sci.* **2022**, *12*, 4984. [[CrossRef](#)]
44. QualiChain JobCrawler Github Repository. Available online: <https://github.com/QualiChain/JobCrawler> (accessed on 10 May 2022).
45. Undergraduate Studies—Undergraduate Education in ECE School. Available online: <https://www.ece.ntua.gr/en/undergraduate/info> (accessed on 10 May 2022).