International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 7, pp. 1723-1736
doi: 10.25147/ijcsr.2017.001.1.128
https://stepacademic.net

Short Paper

Optimizing the University Student Advising Process through Weighted Minimum Spanning Tree Algorithm

Danilo C. Sabado
College of Computing and Information Sciences, University of Makati, Philippines
danilo.sabado@umak.edu.ph
(corresponding author)

Date received: September 19, 2022
Date received in revised form: October 26, 2022; November 28, 2022;
Date accepted: December 5, 2022

Recommended citation:

Sabado, D. C. (2023). Optimizing the University Student Advising Process through Weighted Minimum Spanning Tree Algorithm. *International Journal of Computing Sciences Research, 7,* 1723-1736. https://doi.org/10.25147/ijcsr.2017.001.1.128

Abstract

Purpose – This paper discusses a new approach combining the network graph theory on minimum spanning tree and ensemble-based machine learning algorithm for weight generation to solve an optimization problem for common university advising processes. University advising requires an expert's validation to identify an optimal set of courses that can be recommended to students to finish their program of study at an earliest time while giving heavy considerations on the volume of workload for a given semester.

Method – The study was analyzed utilizing the historical dataset from students' academic profiles from a local university from the Philippines. The process undergoes a set of weight generation, MST model application and ensemble scoring method.

Results – The proposed algorithm was validated using the expert validated advising results which yielded to 92% accuracy.

Conclusion – This paper attempts to develop a solution to the tedious process of student advising through representing the curriculum in a directed graph and implementing a minimum spanning tree algorithm with ML-based weight edges.

Recommendations – To address the potential reasons on the detected accuracies, it is further recommended to include all other constraints such as accommodating co-requisites between courses.

Practical Implications – The proposed algorithm can be implemented in a web-based application or integrated in an enrollment information system for a more justifiable appreciation of the envisioned process improvement.

Keywords – minimum spanning tree, student advising, university, ensemble weighting, optimization

---

INTRODUCTION

Traditional subject recommendation by the faculty members is a critical and complicated activity due to the process of thoroughly investigating individual cases of students, particularly irregular students at most universities like a local, government run university in the Philippines. It inevitably requires human intervention as it requires to pass through decision making based on different factors. One of these factors includes the grades in previous subjects taken, significantly, the prerequisite courses. The decision of a particular student to shift programs or transfer to a different school is another factor that a dean or an equivalent position needs to take into consideration for course advising. Also, the workload and the study hours of courses for a single student is a significant factor for course advising. This process generally takes a substantial amount of time that could only partake for a few concerned students.

This study focuses on the development of an algorithm that can be utilized in the course advising process in a university. Utilizing an ensemble algorithm to assign weights in a network model of minimum spanning tree (MST), this paper aims to identify the optimal set of courses that can be taken by a student for a given semester to finish the degree the earliest possible time without the trade-off for the combination of courses with higher level of difficulty. The MST is utilized to visualize the set of courses in a specific program. It is used for approximating the optimal path like the traveling salesman problem. Since each subject on the map is required to be taken by the student, all nodes should be visited. The goal is to identify the shortest path while visiting all courses. The complexity of this problem can be appreciated for irregular students where there are numerous constraints that need to be considered. These include the maximum number of units that can be taken each semester, the lecture and laboratory hours, and the prerequisite courses. To identify the shortest path, each edge connecting each node should be labeled or assigned by a certain value - the weights.

The proposed algorithm is envisioned to be implemented in a university setting where student advising can be done in a self-help manner, giving them an optimal list, of which courses they should take for the next semester. This can also be used as a source for an early warning or flagging system to students who are at risk of not graduating on time based on the program of study. The system will use machine learning algorithms to analyze the students' grades to perform the aforementioned feature. The system can also evaluate the grades of the students and determine their academic status.

LITERATURE REVIEW

Related Studies

Over the years, there have been several studies that deal with the generalization of optimizing processes in the domain of the logistics on the students' program of the study. In 2013, Mohamed Aly et al., (2013) and colleagues proposed a framework that can be used for student advising processes. Their proposed framework utilizes classification and clustering algorithms. Their work aims to provide a safer educational track that will guide the first-year university students. Unguided students are susceptible to obtaining lower success rates to finish a certain degree if they opt to choose highly demanding courses for specific semesters. Their proposed method was tested using a real case study from "Cairo Higher Institute for Engineering, Computer Science, and Management" using a 12-year student data (Mohamed Aly et al., 2013).

Aldrich (2014) provides an implementation of a directed acyclic graph. In his research, he represented the curriculum of a certain program from a university as a complex system. The courses were visualized as nodes and the prerequisites as the links or the edges of the graph. This concept was heavily adapted in this research. Problems are commonly easier to understand when visualized using certain representations such as tables and graphs. This is beneficially the core concept used in understanding network and graph theories (Mosbah et al., 2017).

Furthermore, the partitioning of the nodes which may be necessary in solving problem domains. Similar to a curriculum where courses are commonly grouped considerably across semesters, clustering nodes in a directed graph can also help visualize the necessary grouping of elements.

Bhavani Likitha Vijjapu (2019) presented a prediction algorithm for major courses selection that uses SVM and Logistic Regression. Their work yielded an overall accuracy of 87% and 90% respectively. Partially similar in terms of problem domain, their research is centered on predicting appropriate choice of major courses but not the entirety of the program.

Ganeshan and Li (2015) move from traditional classification algorithms to collaborative filtering techniques to generate a recommendation for student advising. Directly relative to the problem domain, the recommendation of courses is based on the similarity of the variables contained in the same cluster.

Similarly, Esteban et al. (2020) also utilized collaborative filtering in an approach to help university students in choosing their elective courses. Their study utilized multiple criteria for both students and courses in a genetic algorithm that aims to find an optimal solution. A test instance yielded 84% accuracy of the recommendation.

Behera et al. (2020) developed a rule based automated ML approach that also targets the same approach but on a different problem domain. Their recommender engine targets company performance forecasting when introducing new products or services. Relatively, the decision-making aspect of their project directly translates with the decision support system of the proposed algorithm in this paper.

In a qualitative study of Lynn and Emanuel (2021), the different recommender systems were surveyed and evaluated. The research focused on four major approaches such as Hybrid Recommendation, Knowledge-Based Recommendation, Content-Based Recommender methods and Collaborative filtering (CF). The paper concluded that the hybrid approach works better than the other approaches analyzed.

A similar rule-based approach in developing a recommendation system for student advising was also used in a study of (Saraswathi, 2014). The major drawback with this approach is the need for customizability and require huge effort to be applied in a different setting or environment.

This paper utilizes three different classical machine learning classification algorithms to generate the weights of the courses that can be recommended to students. These algorithms are Support Vector Machine (SVM), C4.5 and Naive Bayes algorithms.
SVM is a supervised machine learning algorithm. The SVM decision function is a better term for an ideal "hyperplane" that is used to split (i.e., "classify") observations into one of many classes based on feature patterns. The most likely label for unknown data may then be determined using that hyperplane (Pisner & Schnyer, 2020).

C4.5 algorithm is a type of decision tree classifier in developing a supervised data mining or machine learning model. Quinlan created Tree C4.5 in the 1990s, based on the Iterative Dichotomiser (ID3) method, which is efficient, powerful, and popular. In general, the C4.5 method consists of two steps: decision tree preparation and rule creation (structure and design). The entropy and information gain with the greatest attribute are then calculated (Budiman et al., 2018).

One of the most popular data mining algorithms is the Naive Bayes, which is based on the Naive Bayes theorem. It calculates the likelihood that a new sample belongs to a particular class based on the premise that all qualities are independent of one another. The necessity to estimate multivariate probabilities from training data has led to this assumption (Chen et al., 2020).

Research Gap

The existing studies that pertain to the utilization of certain network and graph theory in solving optimization problems are very limited. This may be attributed to the customization requirement for each problem domain. To address this problem, this paper aims to design a generic algorithm capable of accommodating different requirements toward finding optimal paths in a directed graph.

The concept will yield the development of a generic algorithm that is capable of performing the recommendations on courses that need to be taken by the student regardless of the curriculum, course, department or college. The variables are designed to be globally and commonly used variables in designing students' programs of study.

METHODOLOGY

The overall algorithmic process of the proposed method follows the structure presented in Figure 1.

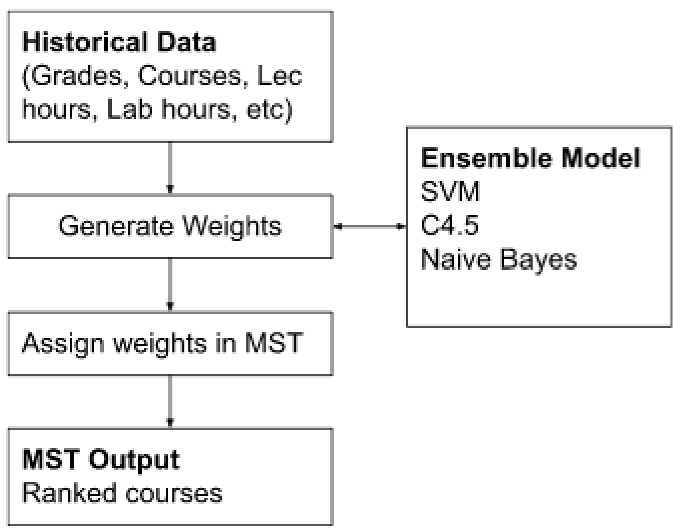

Figure 1. Algorithmic structure of the proposed method

The process starts with gathering relevant historical data for the model. The historical data came from the academic records of students. These serve as quantitative variables that contribute to the generation of weights that will eventually be assigned to the edges of the directed graph.

SVM, C4.5 and Naive Bayes algorithms are used in the ensemble algorithm after performing different simple classification experiments based on the same datasets. Across the different classic traditional algorithms, SVM, C4.5 and Naive Bayes Classification models yielded the top three highest accuracy scores.

Data Collection and Preprocessing

Since not all fields from the raw data are necessary for the development of the model, the fields were trimmed down, and irrelevant ones were deleted. The selected attributes are shown in Table 1.

Table 1. Identified Features for ML-based Weight Modeling

Attribute	Description	Data Type	Range
Course Code	Uniquely identifies a course	Nominal	N/A
Units	Total number of units	Continuous	1-10
Semester	Semester where the course is associated with	Nominal	1-3
Year Level	Year Level where the course is associated with	Nominal	1-5
Grade	Grade earned by a student on a specific course	Continuous	0-100
Lec Hours	Lecture hours for a specific course	Discrete	1-5
Lab Hours	Laboratory hours for a specific course	Discrete	1-5

Fields that were removed from the raw dataset include the course description, learning outcomes and whether a course is a required course by the commission on higher education or not. These fields do not directly affect the assessment approach on what courses are suggested to be taken by the students with his or her current standing. The overall dataset contains 167 historical records which were split into 117 training dataset and 50 validation datasets. This corresponds to a 70-30 training-test split.

Weight Generation

To implement the concept of minimum spanning tree in recommending courses to be taken by the students, it is necessary to provide weights on each edge connecting the nodes or courses. Weights on each edge will be calculated using a machine learning algorithm as proposed in this study.

Presented in Table 2 is the sample representation of how weights were assigned on each course. This is fundamental to the identification of the shortest path, and further suggest courses which are optimally necessary to be taken by a specific student.

Table 2. Sample Subject-Weight Assignment

COURSE CODE	PREREQUISITE	WEIGHT
COMPROG2	COMPROG1	95.00
INFOMAN	COMPROG1	94.00
COMPSTAT	GEMMW, DATANA	88.00
GESTS		87.00
GEPC		87.00
GEFIL2	GEFIL1	83.30
GESSP		81.00
NSTP2	NSTP1	78.70
PE2	PE1	76.40
MODPHY		73.00

Ranked Output

The generation of the weights are based on the ensemble algorithms utilizing the quantitative variables presented in Table 1. To illustrate, Table 3 shows the scores gathered of the available courses not yet taken by the student. Each algorithm scored the courses and their mean scores served as the final basis. The first five highest-ranking courses were selected as a maximum unit and are treated as a constraint that needs to be considered.

Table 3. Ensemble and Final Recommendation Selection

Course	C4.5	SVM	Naive Bayes	Mean Score	Rank
COMPROG2	94.00	95.02	96.04	95.02	1
INFOMAN	93.00	94.29	95.59	94.29	2
COMPSTAT	93.00	93.99	94.98	93.99	3
GESTS	92.00	93.02	94.04	93.02	4
GEPC	91.00	92.01	92.34	91.78	5
GEFIL2	90.00	90.99	91.98	90.99	6
GESSP	90.00	90.97	91.94	90.97	7
NSTP2	85.00	86.04	87.07	86.04	8
PE2	82.00	82.91	83.82	82.91	9
MODPHY	80.00	81.60	81.30	81.22	10

Utilizing a structured dataset (as shown in Table 1), the variables are used as input to the classification models.

Minimum Spanning Tree

The plan or program of study can be represented in a directed graph or network diagram. The nodes represent the courses, and the edges represent the prerequisites or dependencies of each course to one another. The diagram presented in Figure 2 represents the directed graph of the Bachelor of Science in Computational and Data Sciences (BSCDS) curriculum of a local university in the National Capital Region, Philippines.

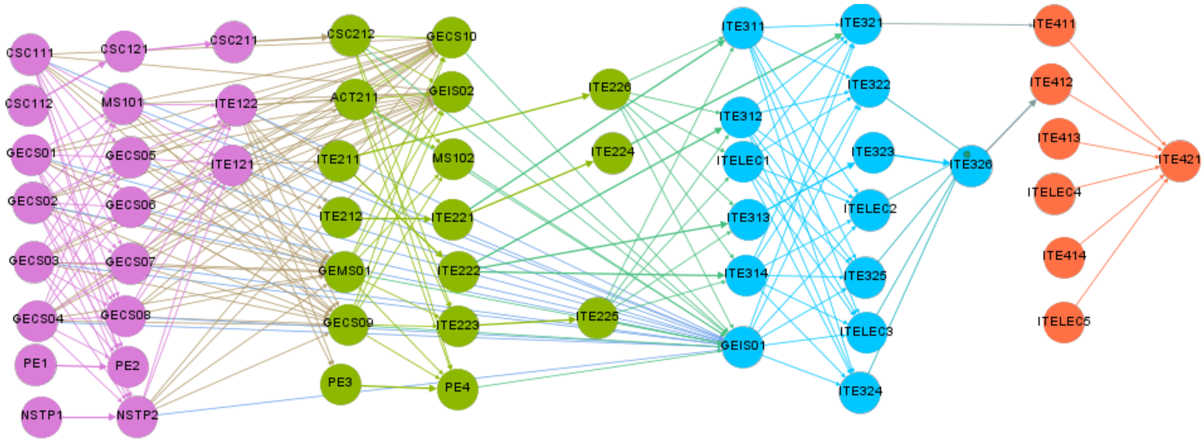

Figure 2. Sample Directed Graph of Prerequisite Mapping for Bachelor of Science in Information Technology Program

The directed graph displays all the courses necessary to be taken by the student to complete the program. The edges across each vertex are assigned with weights computed based on the ensemble driven machine learning algorithms.

Course Recommendation

The courses already taken by the students are automatically eliminated from the set of the possible courses that can be recommended to the student. Thereafter, the remaining courses are ranked based on the computed ML-based weights as presented in Table 3. The prerequisites (see Figure 3) are considered and taken into consideration when choosing the list of courses to be recommended.

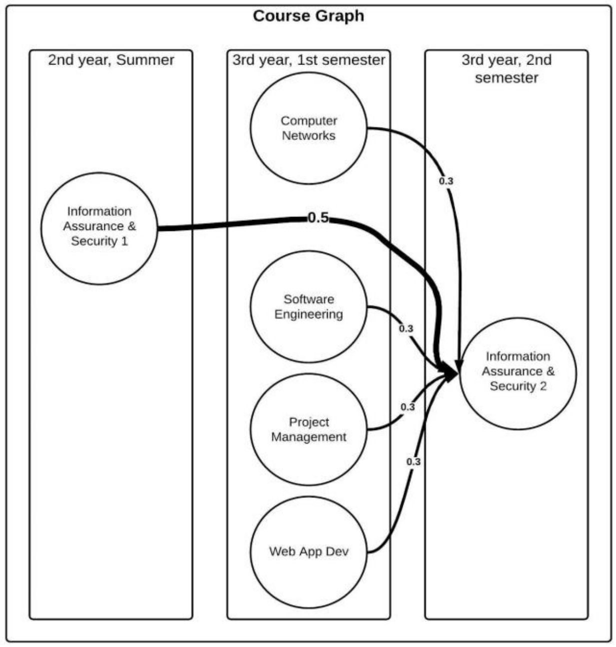

Figure 3. Course Graph Weighing

Though presented visually as seen in Figure 2, the implementation of the MST algorithm is based on Kruskal's Algorithm (Biswas et al., 2016; Khan et al., 2018). This performs the identification of the most optimal path to follow, observing the algorithm flow presented in Algorithm 1.

Algorithm I. Kruskal's Algorithm

```
KRUSKAL(G):
A = ∅
For each vertex v ∈ G.V:
    MAKE-SET(v)
For each edge (u, v) ∈ G.E ordered by increasing order by weight(u, v):
    if FIND-SET(u) ≠ FIND-SET(v):
    A = A ∪ {(u, v)}
    UNION(u, v)
return A
```

Kruskals' algorithm is a known method for finding the minimum spanning tree of a given weighted graph (Ayegba et al., 2020). It takes a graph as an input, finds the subset of the edges that forms a tree which includes every vertex, and has the minimum sum of weights among all the possible formed trees from the graph (Cong & Sbaraglia, 2006).

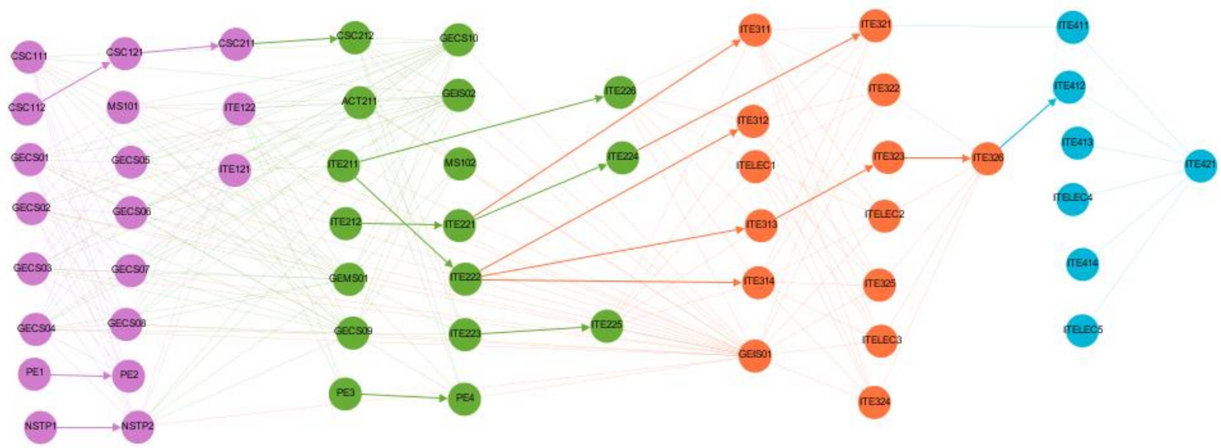

Figure 4. Sample recommended path after undergoing MST

After undergoing Kruskal's algorithm, the model returns a set of courses (set A) that are recommended to be taken by the student based on the input values (student historical grades). This process is reflective of the path highlighted in Figure 4.

Validation

The final model was validated using 50 instances of students' historical records. Data were inputted in the model of varying settings such as regular and irregular students across different year levels. With their already complete historical records, these were used to compare the recommended list of courses to be taken by the students versus what courses were taken by them. This gives an overall impression of the recommendations and the established ground truth. The results of the validation are presented in the next section.

RESULTS

The algorithm was tested using the 50 test instances from historical records of the students in the Computational and Data Sciences and Computer Science programs (Table 4). The dataset is composed of two classifications of records: regular and irregular students.

Table 4. Validation data set distribution

Class	CDS	CS	Total
Regular Students	13	16	29
Irregular Students	8	13	21
TOTAL	21	29	50

The 50 test instances mentioned above were validated by the university personnel and performed the usual advising processes to identify the set of courses that should be taken by each individual student in the roster. Similar dataset (based on the previously discussed train-test split) served as an input to the developed model and a list of recommendations served as an output. The outputs were compared to the expert-validated instances to identify the accuracy of the proposed model.

The result of the validation is presented in Table 5. The straightforward approach of evaluating the students' program of study and what courses to take for the next semester make it effortless for the algorithm to produce accurate recommendations.

Table 5. Validation Result

Class	CDS			CS			
	Instances	Correct	Acc	Instances	Correct	Acc	Total
Regular Students	13	13	100%	16	16	100%	100%
Irregular Students	8	6	75%	13	11	85%	90%
TOTAL	21	19	90%	29	27	93%	

DISCUSSION

For the irregular students, however, there are few deviations with the accuracy based on similarity to the expert's perspective. The resulting values, however, are still acceptable yielding to 90% and 93% accuracy for CDS and CS students respectively.

The misclassified recommendations can be attributed to other rule-based conditions which were not initially considered in the development of the model. An example would be the possibility of enrolling co-requisite courses. This kind of concern can be further implemented and considered in the model development, making the entire model more flexible and adaptive to the custom requirements of a certain university.

CONCLUSIONS AND RECOMMENDATIONS

This paper attempts to develop a solution to the tedious process of student advising through representing the curriculum in a directed graph and implementing a minimum spanning tree algorithm with ML-based weight edges. The algorithm was validated versus the expert's individual evaluation to measure the accuracy of the proposed method. Validating the data across the 50 validation test instances yielded to an overall accuracy of 92%.

To address the potential reasons on the detected accuracies, it is further recommended to include all other constraints such as accommodating co-requisites between courses. To further appreciate the developed algorithm, it is envisioned that the method will be implemented as a supplementary module in a student information management or enrollment system.

LIMITATIONS

This paper centers on the development of a generic algorithm that can be used to assess and evaluate students' historical data to suggest the next set of courses that can be taken. However, every university does have their own custom business rules that directly apply to their assessment and evaluation processes. Hence, the rule-based implementation that varies for each implementation site is not covered by the algorithm. The algorithm only covers the generated set of courses that can be taken optimally based on the historical data.

PRACTICAL IMPLICATIONS

The study is envisioned to be implemented alongside the common student information management systems. This will potentially bring tangible benefits in terms of the expert-required assessment and evaluation processes. Furthermore, the implementation of the algorithm once proven to be accurate and efficient can be used by the students themselves in assessing their standings and understanding their plan of study. This will give them a decision support system level of identifying the optimal combination of courses that can be taken.

ACKNOWLEDGEMENT

The researchers would like to extend their sincerest gratitude to the authors of various works of literature used in this study.

DECLARATIONS

Conflict of Interest

The author declared that he has no conflict of interest.

Informed Consent

No direct private and personal information were used in the conduct of this research.

Ethics Approval

As no private and personal information was used in the research, ethics approval is not necessary.

REFERENCES

Ayegba, P., Ayoola, J., Asani, E., & Okeyinka, A. (2020). A comparative study of Minimal Spanning Tree Algorithms. *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*. https://doi.org/10.1109/icmcecs47690.2020.240900

Behera, R. K., Bala, P. K., & Jain, R. (2020). A rule-based automated machine learning approach in the evaluation of recommender engine. *Benchmarking: An International Journal*, *27*(10), 2721–2757. https://doi.org/10.1108/bij-01-2020-0051

Bhavani Likitha Vijjapu, S. V. (2019). *Machine Learning based Recommendations to aid Educational Planning and Academic Advising through the Virtual Academic Advisor System*. Retrieved November 24, 2022, from http://hdl.handle.net/1773/44137

Biswas, P., Goel, M., Negi, H., & Datta, M. (2016). An Efficient Greedy Minimum Spanning Tree Algorithm Based on Vertex Associative Cycle Detection Method. *Procedia Computer Science*, *92*, 513–519. https://doi.org/10.1016/j.procs.2016.07.376

Budiman, E., Haviluddin, Dengan, N., Kridalaksana, A. H., Wati, M., & Purnawansyah. (2018). Performance of Decision Tree C4.5 Algorithm in Student Academic Evaluation. *Lecture Notes in Electrical Engineering*, 380–389. https://doi.org/10.1007/978-981-10-8276-4_36

Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, 105361. https://doi.org/10.1016/j.knosys.2019.105361

Cong, G., & Sbaraglia, S. (2006). A Study on the Locality Behavior of Minimum Spanning Tree Algorithms. *High Performance Computing - HiPC 2006*, 583–594. https://doi.org/10.1007/11945918_55

Esteban, A., Zafra, A., & Romero, C. (2020). Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization. *Knowledge-Based Systems*, *194*, 105385. https://doi.org/10.1016/j.knosys.2019.105385

Ganeshan, K., & Li, X. (2015). An intelligent student advising system using collaborative filtering. *2015 IEEE Frontiers in Education Conference (FIE)*. https://doi.org/10.1109/fie.2015.7344381

Khan, A., Aesha, A. A., & Sarker, J. (2018). A New Algorithmic Approach to Finding Minimum Spanning Tree. *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*. https://doi.org/10.1109/ceeict.2018.8628095

Lynn, N. D., & Emanuel, A. W. R. (2021). A review on Recommender Systems for course selection in higher education. *IOP Conference Series: Materials Science and Engineering*, *1098*(3), 032039. https://doi.org/10.1088/1757-899x/1098/3/032039

Mohamed Aly, W., Fathy Hegazy, O., & Mohmmed Nagy Rashad, H. (2013). Automated Student Advisory using Machine Learning. *International Journal of Computer Applications*, *81*(19), 19–24. https://doi.org/10.5120/14271-2341

Mosbah, M., Arif, S., Mohammedi, R. D., & Hellal, A. (2017). Optimum dynamic distribution network reconfiguration using minimum spanning tree algorithm. *2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*. https://doi.org/10.1109/icee-b.2017.8192170

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. *Machine Learning*, 101–121. https://doi.org/10.1016/b978-0-12-815739-8.00006-7

Saraswathi, S. S. (2014). Design of an Online Expert System for Career Guidance. *International Journal of Research in Engineering and Technology*, *03*(19), 314–319. https://doi.org/10.15623/ijret.2014.0319057