An automated thesis supervisor allocation process using machine learning

Yuanyuan Fan, Ana Evangelista & Hadi Harb

Engineering Institute of Technology
Perth, Australia

ABSTRACT: The higher education sector has been growing at a significant rate demanding a variety of skills and knowledge from both sides: instructors and learners. Thesis supervision is one of the main challenges as this activity combines research and teaching practices. To ensure the success of a student's thesis project, it is vital to accommodate the student's demands and expectations with the supervisor's availability, experience and knowledge. This article provides an automated process for supervisors' allocation using a machine learning technique based on the current procedure adopted at the Engineering Institute of Technology (EIT), Perth, Australia. The automated process has great potential considering that large numbers of thesis students require supervisors every semester in most institutions. The key to achieve the most suitable student-supervisor match within a short timeframe is assessing certain key factors from both supervisors' and students' sides efficiently. The DecisionTreeClassifier in Python is used for the training of a classification model, as human experience can be translated to a decision tree. The methodology includes the quantifying of supervisor selection criteria, the cleaning of the data, the training and testing of the decision tree model. A case study is conducted to demonstrate the application of the automated process and to validate the efficiency of the automating.

Keywords: Higher education, thesis supervision, machine learning, decision tree

INTRODUCTION

Over the last few years, the supervision of postgraduate students has been widely discussed in literature from different perspectives [1-5]. According to Abiddin et al the roles and responsibilities of supervisors and students must be very clear, which is a critical contributor to thesis completion in a specific period of time [6]. Usually, supervisors are assigned with several tasks, such as teaching, students' guidance and administrative activities, resulting in lack of time to dedicate to thesis supervision. Therefore, the synergy between a supervisor and a student is important to improve the student's chance to succeed. Furthermore, it is common sense that supervisors should hold, at least, the same qualification level of the student's course (degree) [7]. Student-supervisor relationship involves commitment, respect and creativity to mitigate ethical and procedural problems. Considering international students, Filippou et al discuss the roles of thesis supervisors and three supervision modes: teaching, apprenticeship and partnership [2]. The majority of the supervisors play multiple roles, e.g. manager, supporter and critic, depending on their students' abilities and the thesis phase. Given that Master's students lack research experience, the *teaching* mode, among the three supervision modes, is the most usual practice. Considering the supervision-teaching approach, Bruce and Stoodley identified nine categories highlighting the meaning and connection of teaching and learning practices to enhance the research supervision process [8].

Recent studies provide the idea of thesis supervision taking into consideration students' specific needs and demands [9-11]. For instance, some students need extensive support, while others show autonomy to develop their research. To successfully complete a Master's course, it is critical that the level of support from supervisors is considered. According to Maxwell and Smyth …*Supervision is conceptualised here as a complex, creative process resulting in a definite product, a finite dissertation or portfolio, not isolated from the transformative process that produced it* [12]. In addition, these authors indicate three main elements of thesis supervision: the student, the knowledge and the research project [12]. A supervisor's knowledge and expertise are expected to lead the theme of the thesis they supervise and the student's research field. This combination significantly reflects on the supervision effectiveness and the completion of the Master's thesis. Even more, considering blended or virtual learning environments, the accurate match of supervisor knowledge and student research topic reduces major problems, such as poor supervision and student demotivation, consequently impacting the research quality [13][14].

Nowadays, thesis supervision is not only a face-to-face activity. The majority of Master's programmes adopt blended learning or an entirely on-line environment [15][16]. For this scenario, researchers have been exploring new technologies and strategies to improve the connection between supervisors and students. Ahlin and Mozelius provide insights with respects to a synchronous learning environment, e.g. *technology enhances the interaction between instructors and learners* [17]. Therefore, remote modes bring special attention to alternative measures, which can avoid students' sense of isolation, give them possibilities to engage and interact, and encourage them to progress and thrive.

The objective of this article is to propose an automated process to streamline and optimise the allocation of Master's thesis supervisors to respective student candidates. The proposed methodology is based on the current thesis procedure at the Engineering Institute of Technology (EIT), Perth, Australia, where the course coordinator is responsible for assigning suitable supervisors, from industry or academia, to the thesis students. The automated process has been developed using a machine learning technique; more specifically, the DecisionTreeClassifier using scikit-learn in Python [18][19].

Machine learning within the area of higher education has been discussed extensively in other studies [19-23], but not for the thesis supervisor allocation process presented in this study. The proposed automated process becomes more effective and complex as the size of the institution increases and the numbers of students grow.

Next in the article, the authors introduce the thesis procedure at the EIT to set the context of the research work. Then, the research methodology of the automated thesis allocation process using machine learning is proposed. Additionally, the proposed methodology employing the decision tree algorithm is implemented utilising libraries and built-in functions in Python. A case study is included to further explain how to apply the automated process. Finally, the conclusions of the research and future work are presented.

PROCEDURE OF THESIS PROJECTS

In this section, the general procedure of the Master's thesis unit at the EIT is presented, given that the thesis allocation methodology proposed in the article is based on this procedure. However, it is to be noted that the proposed automated methodology is applicable to other institutions as well, with minor modifications to adapt to specific procedures of an institution.

Manual Thesis Supervisor Allocation

The thesis project at the EIT is part of course-work based engineering Master's programmes, as a unit. The thesis unit runs over one semester starting with students submitting a research topic. Upon receiving the thesis topic from each student, the course coordinator needs to allocate thesis supervisors based on a list of factors. It is to be noted that unlike in universities, where students find their own supervisors based on their own interests, the EIT students (studying on-line and on-campus) are based all over the world and they do not have close contact with the EIT academic staff. Thesis supervisor allocation is therefore critical for the EIT students' success in completion of the thesis unit.

As a matter of fact, the supervisor allocation process is equally critical for university students. There are many occasions, when a university student approaches a professor in their university with a proposed thesis topic, but ends up being rejected for various reasons. Some typical reasons of rejection include: the professor is not a specialist in the student's proposed topic, the professor is not familiar with the software the student plans to use, the professor has no capacity to supervise more students than he/she already has, etc. At the EIT, the course coordinator needs to carefully assess each thesis student, so that the most suitable supervisor for each student is allocated.

The thesis allocation process used to be manual at the EIT. The course coordinator would look at all the submitted thesis topics and a list of potential thesis supervisors to match each thesis student to a supervisor. This manual process used to be manageable when there were not many students. As the EIT student number grows, manually allocating thesis supervisors has become tedious and inefficient. The consequences of allocating a less suitable thesis supervisor could be that the student has to change his/her thesis topic as the allocated supervisor is not familiar with the student's initially proposed topic, the supervisor is not able to provide sufficient guidance to the student due to lack of communication means, etc.

Example of Manual Supervisor Allocation

The inefficiency and inaccuracy of manually allocating thesis supervisors are further explained with an example. Suppose there are 20 thesis students. The course coordinator looks at the first thesis topic and selects a supervisor from the potential supervisor list for the student, but later notices that the already selected supervisor is more suitable to supervise another thesis student down the student list. Then, the course coordinator has to go back to reallocate a different supervisor for the first thesis student. Eventually, when all the 20 thesis students have been allocated supervisors by the course coordinator checking back and forth, there could have been tens of times of reallocating. Even with all the time and effort spent, there is still no guarantee that all the 20 thesis students would end up with the best matching supervisors. It has to be noted that in this manual allocation process, the course coordinator needs to consider all the affecting factors relying on his/her strong memory. The affecting factors including time availability, specialised areas, etc, are identified in the methodology section. The unmanageable work load requires an automated process to improve the accuracy and efficiency of thesis supervisor allocation, and therefore ensure the quality of thesis students' research work.

Thesis supervisor allocation is a key factor that determines the success or progress of students' thesis work, especially as most students who are enrolled in coursework-based Master's programmes do not have prior research experience.

METHODOLOGY- APPLYING MACHINE LEARNING IN THESIS SUPERVISOR ALLOCATION

To allocate the most suitable supervisors for each thesis student, i.e. to find the best supervisor-student matches, the idea of automating the thesis supervisor allocation process naturally comes in place.

It is clear that allocating thesis supervisors is not a mechanical process. Therefore, an intelligent algorithm needs to be applied to automate the process. The common and best received solution to develop a smart system or process is to apply machine learning. As it is known, machine learning is a data driven approach to let the machine make decisions like humans based on what it learns from existing data [23]. So, there needs to be adequate data to apply a machine learning algorithm, either as classification or regression. However, the thesis supervisor allocation work is qualitative, which makes applying machine learning challenging. The key is thus to convert the qualitative information into quantitative representatives. One of the methods to do so is using the concept of knowledge representation in machine learning [24].

To automate the procedure of matching supervisors with students, a classification problem is defined. Selection criteria of supervisors are identified based on the course coordinator's knowledge and experience. The aim of the classification solution should be to select supervisors from a potential supervisor list using the identified selection criteria as features. The supervisors naturally are class labels of the classification problem.

In this article, the decision tree classifier is used as the classification algorithm because it is easy for knowledge and experience interpretation. A binary decision tree consists of a root node, internal nodes and leaf nodes containing class labels, as show in Figure 1. A feature is assessed, and a decision is made from the root node down to the branches and internal nodes until a leaf node is reached, which would be the final decision of a classification. At the root node and each internal node, the decision to follow one of the two subsequent paths is based on the evaluation of a feature as either *1* or *0*.

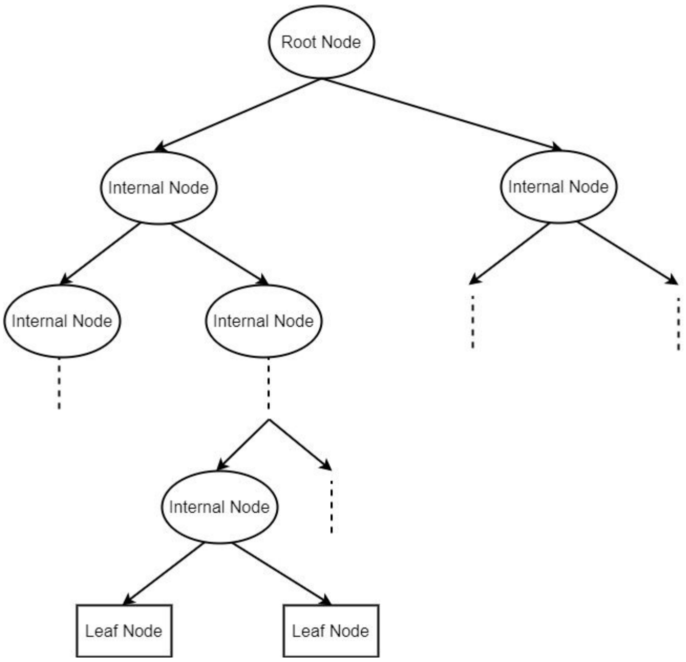


Figure 1: Schematic diagram of a decision tree.

The manual thesis allocation process could be presented as an application of a decision tree algorithm as shown below:

1. The course coordinator selects an important feature according to him/her.
2. Only supervisors for whom the feature is true are retained.
3. Another feature is selected.
4. Steps 2 and 3 are repeated until a stopping criterion is reached, such as when only one supervisor remains.

It is understandable that with the same list of features, the resulting structure of the decision tree is different if the features are assessed in different orders. Therefore, it is important that the features are evaluated in a proper sequence, such that the data is separated in the best way possible at each decision step. The best sequence should result in the minimum number of decision steps [18]. The decision tree algorithm optimises the process by minimising the number of steps, i.e. at each step selecting the best feature to split the data. To achieve this, the decision tree algorithm employs the concept of *impurity*. Ideally, after each splitting, the subsequent decisions are only made based on the unused features, which means the impurity of the splitting is always *0*. There are different metrics that the decision tree algorithm can use to achieve the lowest impurity possible [18].

After the features and labels are defined, the data to train a decision tree needs to be extracted from existing information. The way to produce the attribute data is to assign values of *0* and *1* by comparing each supervisor against each attribute or feature. The value of *1*, indicating *yes*, is assigned when a supervisor matches an attribute or selection criterion. Naturally, the value of *0* is assigned when a supervisor does not match an attribute. However, at times, the decision of assigning *0* or *1* is not clear in the quantifying process. In other words, both *0* and *1* can be acceptable. In this case, the assigned values are *0* or *1*. All the three conditions will be elaborated in the implementation section. Once all the potential supervisors are assessed based on the attributes, the rows having the same attribute values with any previous row will be removed to avoid duplication. A supervisor assigned to a removed row is then grouped to the supervisor in the previous row who has the same attribute values. Eventually, the class labels in the last column represent numbered supervisor groups rather than individual supervisors. The supervisors in each group have the same attributes. Therefore, the produced data will be in a pattern as shown in Table 1.

Table 1: Pattern of data produced - attributes and labels.

Attribute - 1	Attribute - 2	…	Attribute - *n*	Label
1	1	…	0	Supervisor - 1
0	1/0		1	Supervisor - 2
…	…	…	….	…
1	0		1/0	Supervisor - *n*

In the binary decision tree outlined in the article, the attribute values can only be *1* or *0*. Therefore, the rows with attribute values of *1/0* need to be split into multiple rows, so that the DecisionTreeClassifier in Python can process the data. The way to do this is to take *1/0* as *1* or *0*. In this case, if a row has n number of *1/0*, the expanded result of this row would be 2^n number of rows with *1*'s and *0*'s only. It is possible that duplicated attribute rows occur after the data expansion. So, it is important to clean the data by removing duplications before use. In Python, this can be done using a built-in method to keep the uniqueness of an entire or part of a data frame. For this research, the uniqueness of the data is confined to the attributes only. The labels of the rows with identical attribute values are still different. In other words, for the same student, more than one supervisor could be suitable to supervise him/her. This gives rise to the question of which unique row to keep. In Python, the method to drop duplicate rows provides the option of keeping the first or last instance of any duplicated rows. As far as this research is concerned, keeping the last or first instance has essentially the same meaning given that the supervisor groups are numbered at an arbitrary sequence. Therefore, either keeping the first instance or the last instance can be applied.

The cleaned data is then split into training data and testing data to achieve a decision tree model and to test the accuracy of the achieved model. The data splitting process can be randomly selecting a certain number of rows of the data frame as the training data and using the rest of the rows as the testing data. However, for this research, randomised data splitting may end up excluding important instances from the training data set. This would lead to the localisation of the resulting decision tree model, which will not have a good accuracy score on the testing data. For example, if all rows of Supervisor - 1 fall into the testing data set after the splitting, then it is equivalent to that the Supervisor - 1 group does not participate in the decision tree training. This of course misses out important information to achieve an accurate model. Therefore, the data splitting step will be done in a way that all data classes participate in model training.

It is worth mentioning that due to the qualitative nature of the research problem, it is impossible to ensure each quantifying step is 100% accurate. Therefore, the machine learning based methodology is strictly speaking a guideline for the course coordinator to allocate supervisors to thesis students. However, the model has shown to be accurate for all the cases (past and new) applied so far. A detailed case study of applying the methodology to a past thesis student is discussed next in this article.

APPLYING MACHINE LEARNING IN THESIS SUPERVISOR ALLOCATION - IMPLEMENTATION

In this section, the automation methodology of thesis supervisor allocation is implemented in the context of the EIT thesis procedure.

Selection Criteria of Thesis Supervisors

Depending on different institutions, the selection criteria of thesis supervisors should be identified on a case by case basis. As mentioned in this article, the authors have used the Master's thesis unit at the EIT to study the implementation of the automated methodology.

The identified selection criteria of the EIT's thesis supervisors are listed in Table 2. The related aspects of each selection criterion are also listed in the table.

Table 2: Thesis supervisor - selection criteria.

Selection criterion	Related aspect
1) Flexible time	Supervisor-student meeting
2) Electrical engineering	Supervisor's academic background
3) Industrial automation	Supervisor's academic background
4) Industry guidance	Supervisor's industry experience
5) Detailed guidance	Supervision experience
6) On-line mode	Supervisor's availability
7) On-campus mode	Supervisor's availability
8) Intelligent algorithms	Supervisor's knowledge
9) Power systems	Supervisor's research area
10) Industry 4.0	Supervisor's research area
11) Instrumentation	Supervisor's specialisation
12) Process control	Supervisor's specialisation
13) Renewable power	Supervisor's specialisation
14) Data communication	Supervisor's specialisation
15) Substation	Supervisor's specialisation
16) Energy storage	Supervisor's specialisation
17) Power quality	Supervisor's specialisation
18) System control	Supervisor's specialisation
19) Power electronics	Supervisor's specialisation

It is to be noted that the sequence of the selection criteria in Table 2 is for the purpose of explanation only. As mentioned in the methodology section, the sequence of the features is determined by the decision tree algorithm using certain impurity metric such that the data are split in the best way possible. In Python, there is a parameter in the DecisionTreeClassifier called *criterion*, which measures the quality of the splitting. Different impurity metrics can be adopted for the *criterion* parameter. In this research, a very common measure Gini impurity index is used.

The formula of Gini impurity index is given below in (1):

$$I_{Gini}(j) = \sum_i p(i|j)(1 - p(i|j)) \tag{1}$$

where *i* is an index associated with each class, *j* is a certain node, and *p*(i|j) is the ratio between the number of samples belonging to class *i* and the total number of samples belonging to the selected node [18]. The values in Equation (1) for the thesis allocation methodology will not be stationary as time goes by because the decision tree will be dynamic eventually. As an example, new specialisations could be added or removed from Table 2 in the future given the rapid development of emerging technologies.

The roles that the 19 selection criteria play in supervisor allocation are elaborated further for the purpose of classification in the following paragraphs. There are three classifications in total, i.e. *yes*, *no* and *possible*, depending on if or how a selection criterion is satisfied by a supervisor.

1) Flexible time - at the EIT, the majority of thesis supervisors are contract-based. This means many supervisors are only available to guide students during after-work hours of their time zones, e.g. the supervisor-student meetings need to be scheduled with great restrictions. For these supervisors, the classification of *flexible time* is *no*. For supervisors who are working full-time with the EIT or are not working full-time with any company or institution, there is more flexibility to interact with their students through meetings. Therefore, these supervisors are classified as *yes* against the *flexible time* criterion. In the case when a supervisor has a part-time workload, the classification is *possible*. The reason is that satisfactory interaction between the supervisor and the student is achievable, but administrative difficulty needs to be addressed.

2) Electrical engineering and 3) industrial automation - all thesis supervisors at the EIT must hold a PhD degree in an engineering field. If a supervisor is a PhD in electrical engineering, then the classification for the supervisor against the *electrical engineering* criterion is *yes*. If a supervisor does not hold a PhD degree in electrical engineering, but is a PhD in a related field, such as power electronics, and holds a Master's degree in electrical engineering, then the classification should be *possible*. This is because the essence of thesis supervision is research guidance. The supervisor should ideally have the highest qualification in electrical engineering, but not necessarily. When a supervisor does not suit either qualification condition mentioned here, the classification is *no*. With the *industrial automation* criterion, the classification principle is the same, given that *industrial automation* is also an engineering field.

4) Industry guidance - if a supervisor has industry working experience in the electrical and/or automation fields, the student tends to receive more practical, rather than theoretical suggestions from the supervisor. This is when the classification for the supervisor against the *industry guidance* criterion is noted as *yes*. When a supervisor has no relevant industry experience, the classification is naturally *no*. In the case, when a supervisor has collaborations with industry from an academic environment, the supervisor can provide industry guidance to some extent and is thus classified as *possible*.

5) Detailed guidance - it has been noticed based on the course coordinator's experience that some thesis students require more detailed guidance than others, due to lack of research experience or other required skill set. For example, some students do not know how to write an abstract of a thesis. For these students, a novice supervisor is preferred, as senior supervisors tend to assume high research abilities from their students and miss certain important details. Therefore, for the *detailed guidance* criterion, a supervisor who has supervised five or fewer students is classified as *yes*, a supervisor who has supervised more than 10 students is classified as *no*, and other supervisors are classified as *possible*.

6) On-line mode and 7) on-campus mode - these two selection criteria relate to supervisors' time availability and time zones. There are both on-line and on-campus thesis students at the EIT. On-campus students have a slightly more condensed thesis semester and shorter thesis duration, compared to on-line students. The general supervisor-student meeting frequencies are weekly for the on-campus mode and fortnightly for the on-line mode. When a supervisor is available to meet the student once a week, the classification against the *on-line mode* and *on-campus mode* criteria are both *yes*; when the meeting frequency can only be twice a month with a supervisor, the classification is *yes* for *on-line mode* and *no* for *on-campus mode*. Besides, on-campus students prefer face-to-face meetings with their supervisors. So, to reserve local supervisors for on-campus students, a supervisor is classified as *no* for on-line and *yes* for on-campus when the supervisor is locally based.

For the *on-campus mode* criterion, sometimes a supervisor cannot have formal weekly meetings with the student, but can maintain the student's progress through other means of communication, e.g. mobile, with the student. This is when the classification of *possible* applies. The *on-line mode* criterion usually does not have the classification of *possible*. This is because all supervisors can manage on-line supervision meetings as long as scheduling allows, given that on-line supervision availability is the minimum administrative requirement when supervisors are recruited.

8) Intelligent algorithms - intelligent algorithms, such as neural networks, partial swarm optimisation, etc, can be applied to many aspects in the industrial automation and electrical engineering fields. Therefore, supervisors' in-depth knowledge in intelligent algorithms is beneficial to students' thesis work, especially when control design is involved in a thesis topic. If a supervisor's research scope does not include the application of intelligent algorithms, the supervisor is classified as *no*. If a supervisor has the knowledge of intelligent algorithms, but is not familiar with the relevant tools, e.g. the MATLAB machine learning toolbox, the classification is *possible*. The reason is that the familiarity of the algorithm tools from the supervisor's side can assist the students with their research, but is not essential in thesis supervision. Naturally, when a supervisor is an expert in intelligent algorithms, the classification is *yes*.

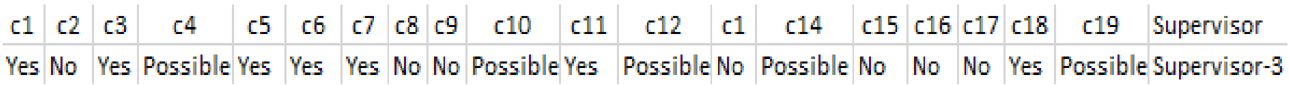
9) Power systems and 10) industry 4.0 - both *power systems* and *industry 4.0* are broad research areas that distinguish supervisors working in different fields. If a supervisor's major research area matches an area, the classification is *yes*. If a supervisor has a research background focusing on automation and data communication in many engineering fields including power systems, then the classifications for the *power systems* and *industry 4.0* criteria are *possible* and *yes*. If a supervisor's major area is *power systems*, but has a focus on automation and data communication, then the classification for the *power systems* and *industry 4.0* criteria are *yes* and *possible*. The classification is obviously *no* for both research area criteria, if a supervisor does not work in the area at all.

11) Instrumentation to 19) power electronics - these selection criteria are all specialised sub-areas. Taking *energy storage* as an example, if a supervisor's research interest is in energy storage related systems, the classification for the criterion is apparently *yes*. If a supervisor deals with a particular type of energy storage systems, e.g. battery storage, the classification for the supervisor is *possible*, given that different types of storage systems have similar principles. If a supervisor works in power systems, but does not have any experience in energy storage, the classification is *no*. The same classification principle applies to other selection criteria regarding the supervisor's specialisation aspect.

Selection Criteria Interpretation and Data Cleaning

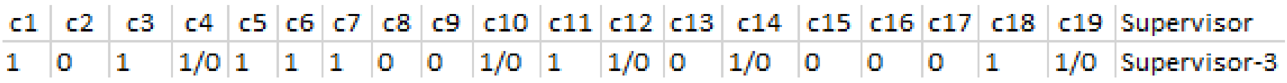
To automate the thesis supervisor allocation process, a list of 18 groups of potential supervisors was evaluated and classified into three categories against each selection criterion as mentioned earlier. To apply machine learning, the classifications must be digitalised. For this, the classifications of *yes*, *no*, and *possible* are indicated by *1*, *0* and *1* or *0*, respectively.

The digital interpretation of the classifications, using Supervisor - 3 as an example, is shown in Figure 2, where the 19 selection criteria in Table 2 are indicated by *c1* to *c19* in sequence for simplicity. The classifications after digitalisation become attribute values as shown in Figure 3.



c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c1	c14	c15	c16	c17	c18	c19	Supervisor
Yes	No	Yes	Possible	Yes	Yes	Yes	No	No	Possible	Yes	Possible	No	Possible	No	No	No	Yes	Possible	Supervisor-3

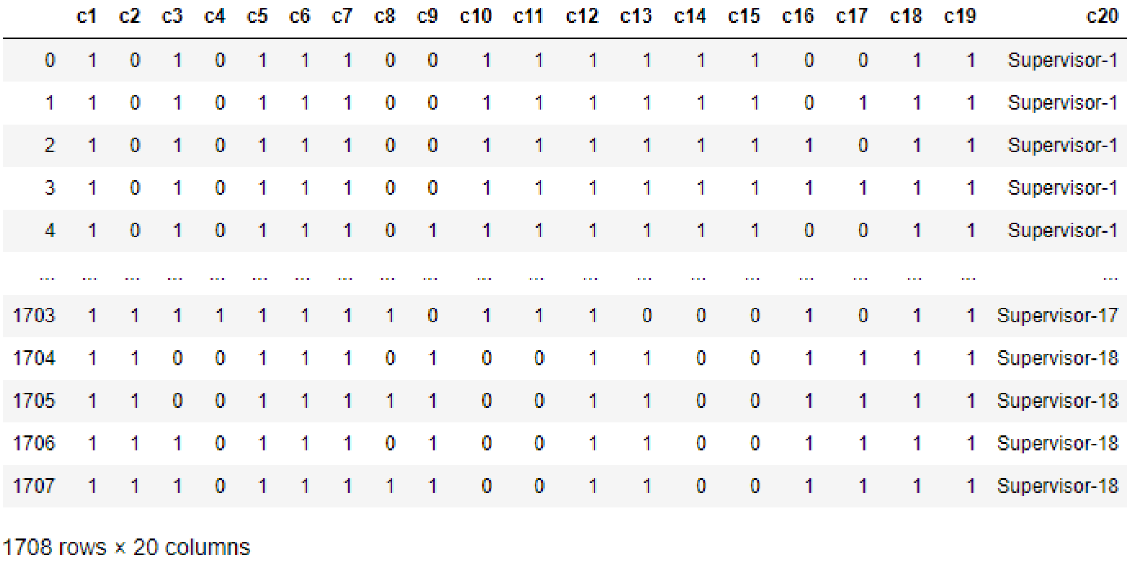
Figure 2: Example of supervisor classifications before digitalisation.



c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	Supervisor
1	0	1	1/0	1	1	1	0	0	1/0	1	1/0	0	1/0	0	0	0	1	1/0	Supervisor-3

Figure 3: Example of supervisor classifications after digitalisation.

As mentioned earlier, the DecisionTreeClassifier implementing scikit-learn in Python is used to train a model and select supervisors. The data as shown in Figure 3 needs to be further interpreted. Still using Supervisor - 3 as an example, there are altogether five *1/0* attribute values. This means the one row of Supervisor - 3 in Figure 3 is converted to 32 (2^5) rows. After interpreting all the classifications of the 18 supervisor groups, the rows of the entire table or data frame are added up to 1,708, as shown in Figure 4. It is noted that the head of the *supervisor* column, i.e. the label column is marked as *c20* for ease of programming.

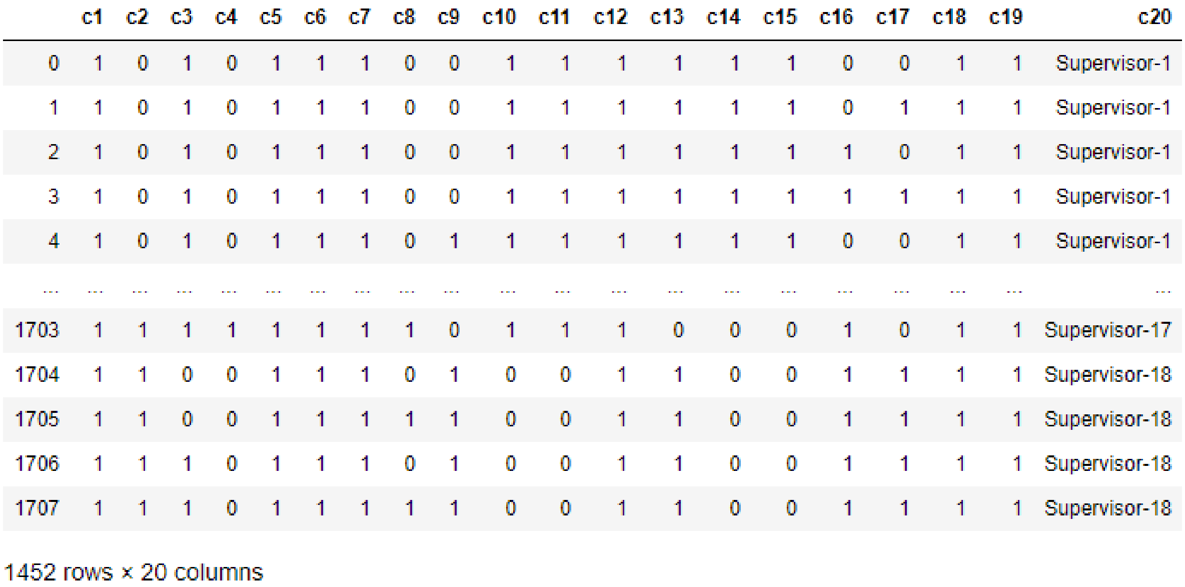


	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
0	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	Supervisor-1
1	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	Supervisor-1
2	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	Supervisor-1
3	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	Supervisor-1
4	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	Supervisor-1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1703	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	Supervisor-17
1704	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18
1705	1	1	0	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18
1706	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18
1707	1	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18

1708 rows × 20 columns

Figure 4: Data frame of supervisor classifications after interpretation.

Now the next step is to keep the rows with unique attribute values only, i.e. to remove any repeating rows which are identical from Column *c1* to Column *c19*. For this, the *drop_duplicates* method in Python is applied to clean the data. In the *drop_duplicates* method, the parameter of *keep* is assigned as *last* to keep only the last instances when there are repeated attribute rows. The number of rows of the data frame after removing duplicates is reduced to 1,452, as shown in Figure 5.

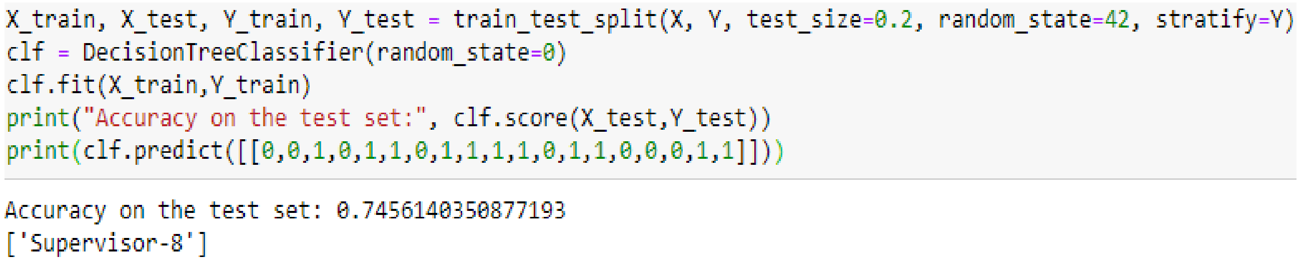


	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
0	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	Supervisor-1
1	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	Supervisor-1
2	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	Supervisor-1
3	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	Supervisor-1
4	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	Supervisor-1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1703	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	Supervisor-17
1704	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18
1705	1	1	0	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18
1706	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18
1707	1	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	Supervisor-18

1452 rows × 20 columns

Figure 5: Data frame of supervisor classifications after dropping duplicates.

As the purpose of this research is to determine a supervisor for a given thesis student, the *supervisor* column, i.e. *c20* is the class label and the rest of the columns *c1* to *c19* are the features. To fit the model, the 19 attribute columns are denoted by *X* and the labels are denoted by *Y*. The training and testing data are split in a ratio of 80% to 20%. To avoid missing information, the parameter of *stratify* is used to ensure the same splitting ratio is applied to each class (label or supervisor group). Also, given that the data sizes of each class are comparable, the accuracy score of the model on the testing data set is used as the metric to evaluate the decision tree. Figure 6 shows the code using the *DecisionTreeClassifier* in Python. The criterion parameter in the classifier is not explicitly shown in the code because when no function is assigned to *criterion*, the Gini impurity is selected by default. The *.score* method used in the program returns the mean accuracy on the testing data. The accuracy score is 0.75 as shown in Figure 6, which is a good score for a classification model. A list of random inputs is used to provide a visual understanding of the model and a supervisor (in Group *Supervisor - 8*) is predicted.



```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42, stratify=Y)
clf = DecisionTreeClassifier(random_state=0)
clf.fit(X_train,Y_train)
print("Accuracy on the test set:", clf.score(X_test,Y_test))
print(clf.predict([[0,0,1,0,1,1,0,1,1,1,1,0,1,1,0,0,0,1,1]]))
```

```
Accuracy on the test set: 0.7456140350877193
['Supervisor-8']
```

Figure 6: Modelling using the DecisionTreeClassifier.

To have a better visualisation of the trained decision tree, it is worth to draw the resulting tree. However, given the large amount of data of the problem, it is challenging to have a clear vision of the decision tree. Figure 7 shows the complexity of the decision tree drawn using Python.

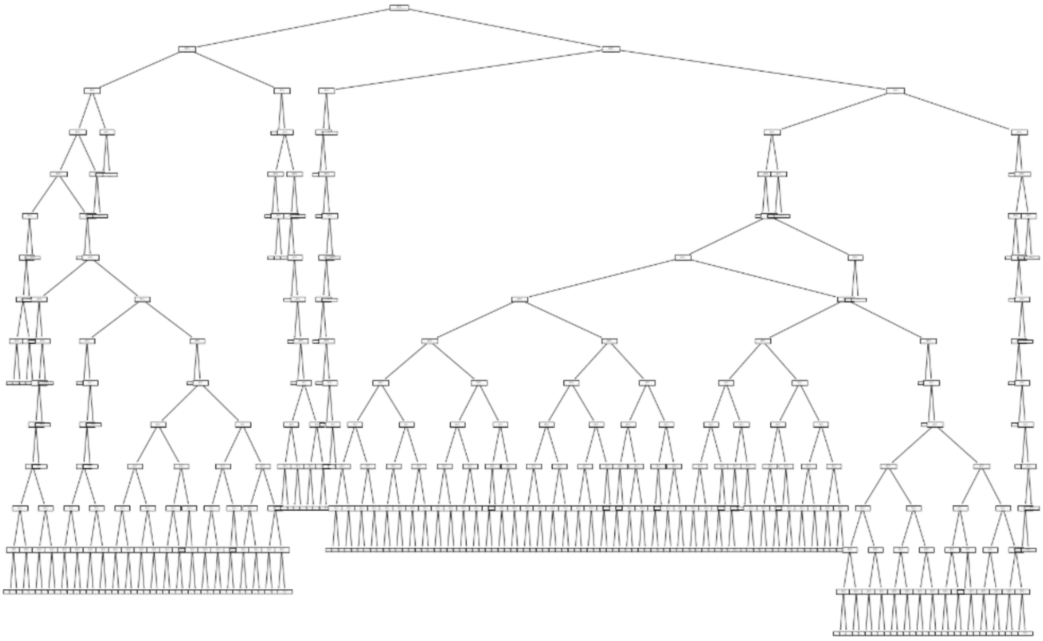


Figure 7: Trained decision tree.

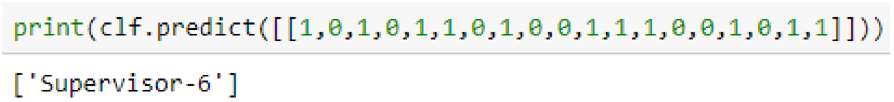
CASE STUDY

It is obvious that to allocate a supervisor for a thesis student, the 19 attributes or selection criteria need to be carefully studied and the values need to be determined. This is done by assessing the student's specific condition. When a thesis student requires the satisfaction of a supervisor against a selection criterion, the attribute value for that criterion should be *1*; otherwise, it is *0*. To clearly demonstrate and verify the automated supervisor allocation process in detail, a real thesis student example is used as a case study in this section.

The student was an on-line student and has already completed his thesis under the guidance of his supervisor. To avoid any sensitive information, the actual thesis title is not presented, but is reworded as *Cost analysis and optimisation of PV systems - a case study*, keeping the essence of the thesis topic. To determine the attribute values of the case for the decision tree model, the 19 selection criteria are assessed as follows:

1) Flexible-time - the student was a full-time employee and could only meet after work hours. This means a supervisor with flexible time availability was preferred, which gives *1* as the value for the attribute of *flexible time*.
2) Electrical engineering - the key words in the thesis topic were *analysis* and *optimisation*, so a supervisor with an electrical engineering background was not required. This gives *0* as the attribute value for *electrical engineering*.
3) Industrial automation - the key word *optimisation* in the thesis title is related to *control*. Therefore, the attribute value for *industrial automation* is *1*.
4) Industry guidance - the student was working in the photovoltaic (PV) sector at the time of conducting his thesis, so he had industry experience and did not need industry guidance from his supervisor. Therefore, the value for the attribute of *industry experience* is *0*.
5) Detailed guidance - the student had no prior research experience when he started his Master's thesis unit, and thus required detailed and close guidance from his supervisor. This gives the value *1* for *detailed guidance*.
6) On-line mode - the student was studying on-line, so would need a supervisor that was available for the on-line mode supervision. The attribute value for *on-line mode* is naturally *1*.
7) On-campus mode - the attribute value against the *on-campus mode* is naturally *0*.
8) Intelligent algorithms - the thesis topic was on *optimisation*, which would require the application of *intelligent algorithms* to achieve the best research outcome. Therefore, the attribute value here is *1*.
9) Power systems - based on the thesis topic, the thesis was not directly related to power systems, although power system analysis, e.g. PV system integration might be recommended as future work of the thesis. Therefore, the value against *power systems* is *0*.
10) Industry 4.0 - the thesis topic did not require the technologies related to *industry 4.0*, so the attribute value here is *0*.
11) Instrumentation - the knowledge of *instrumentation* would be beneficial to the thesis work given that *cost* needed to be considered as the research objective. Therefore, the attribute value for *instrumentation* is determined as *1*.
12) Process control - similar to the analysis against the *instrumentation* criterion, *process control* would be beneficial, so the attribute value here is *1*.
13) Renewable power - PV systems are the most important means of solar power generation, so the value is *1* for *renewable power*.
14) Data communication - the knowledge in data communication was not essential considering the thesis topic, so the attribute value here is *0*.
15) Substation - the topic was not related to substation, so the attribute value is *0*.
16) Energy storage - PV is tightly connected to battery storage, so the value for *energy storage* is *1*.
17) Power quality - the value for *power quality* is *0*, as the scope of the thesis topic did not include power quality.
18) System control - the knowledge of system control would be beneficial as cost is directly related to the design of control systems. Therefore, the attribute value of *1* is used here.
19) Power electronics - the knowledge of power electronics is essential as power conversion is impossible without power electronics. This gives the value *1* against *power electronics*.

To summarise the analysis above, the input feature values for the thesis student is [1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1]. Substituting this input in the code, one supervisor in Supervisor - 6 is selected, as shown in Figure 8.



```
print(clf.predict([[1,0,1,0,1,1,0,1,0,0,1,1,1,0,0,1,0,1,1]]))

['Supervisor-6']
```

Figure 8: Supervisor allocation result for a thesis student.

It is worth to be mentioned that the supervisor of the student in the case study was allocated manually by the course coordinator. It was one of the supervisors in Group *Supervisor - 6* that supervised the student who completed his thesis with a *distinction*. The automated process of allocating thesis supervisors has been developed recently. It has only been applied to students who started their theses this semester, i.e. at the time of this article's submission. Therefore, more data to validate the effectiveness of the automated process will be available as more thesis students complete their thesis unit.

CONCLUSIONS

In this article, the authors present the development of an automated process to allocate thesis supervisors using a machine learning tool. When allocating a supervisor for a thesis student, the matching of the thesis topic and the supervisor's expertise is without any doubt the most critical aspect to consider. However, there are other factors that are also important in assigning a suitable supervisor to a student. The assessing aspects or factors vary from institution to institution. In the context of the EIT's thesis procedure, 19 selection criteria are identified to achieve the best matching possible between thesis students and supervisors. The qualitative information is converted to quantitative data, so that machine learning can be applied to automate the thesis allocation process. Regarding this, in conjunction to the human experience represented by the course coordinator, a classification problem is defined and the decision tree algorithm is used to process the data. The DecisionTreeClassifier implementing scikit-learn in Python is selected as the data modelling tool. Data sorting and cleaning are also conducted in the data preparation stage. A decision tree model is developed using part of the data as training data and is tested using the rest of the data as testing data. A case study is conducted to show the application details of the automated process and to further verify the trained decision tree model.

It is to be noted that human judgement and intervention are still needed when the automated process produces unrealistic results. For example, a supervisor may not have enough capacity to supervise more than three thesis students at a time. Nevertheless, the automated process definitely makes the supervisor allocation procedure more efficient and accurate.

It can be seen that the research methodology proposed in the article includes both information qualifying and data analysis. The qualifying process is based on the course coordinator's knowledge, experience and resources. The selection criteria, however, will be undergoing changes for various reasons. For example, thesis topics in a few years may have different research focuses. For the same reason, thesis supervisor groups could be updated as well. Therefore, with dynamic data, the decision tree trained in the article will be a dynamic tree. It is critical that one understands the automated procedure and allows flexibility of the application under different circumstances.

Future work for this project includes using other classification algorithms, such as random forest to compare the results with that of using decision tree. Furthermore, fuzzy logic can be used based on *degrees of truth* rather than the usual *true or false* (*1* or *0*) indicators.

ACKNOWLEDGEMENTS

The authors thank the Engineering Institute of Technology for the information support.

REFERENCES

1. Bazrafkan, L., Yousefy, A., Amini, M. and Yamani, N., The journey of thesis supervisors from novice to expert: a grounded theory study. *BMC Medical Educ*., 19, **1**, 1-12 (2019).
2. Filippou, K., Kallo, J. and Mikkilä-Erdmann, M., Supervising master's theses in international master's degree programmes: roles, responsibilities and models. *Teaching in Higher Educ*., 26, **1**, 82-96 (2021).
3. Amundsena, C. and McAlpine, L., *Learning supervision*: trial by fire. *Innovation in Educ. and Teaching Inter*., 46, **3**, 331-342 (2009).
4. Carter, S., Supervision learning as conceptual threshold crossing: when supervision gets *medieval*. *Higher Educ. Research & Develop*., 35, **6**, 1139-1152 (2016).
5. Middleton, A., Ortiz-Catalan, M. and Gustafsson, M., Supervision of M.Sc. theses using the writing of a scientific article as a framework to increase efficiency and quality of research outcomes. *Proc. 2019 41st Annual Inter. Conf. of the IEEE Engng. in Medicine and Biology Society (EMBC)* (2019).
6. Abiddin, N.Z., Hassan, A. and Ahmad, A.R., Research student supervision: an approach to good supervisory practice. *The Open Educ. J*., 2, 11-16 (2009).
7. Moses, I., Supervision of higher degree students - problem areas and possible solutions. *Higher Educ. Research &. Develop.*, 3, **2**, 153-165 (1984).
8. Bruce, C. and Stoodley, I., Experiencing higher degree research supervision as teaching. *Studies in Higher Educ*., 38, **2**, 226-241 (2013).
9. de Kleijn, R.A.M., Meijer, P.C., Brekelmans, M. and Pilot, A., Adaptive research supervision: exploring expert thesis supervisors' practical knowledge. *Higher Educ. Research & Develop.*, 34, **1**, 117-130 (2015).
10. Vereijken, M.W.C., van der Rijst, R.M., van Driel, J.H. and Dekker, F.W., Novice supervisors' practices and dilemmatic space in supervision of student research projects. *Teaching in Higher Educ.*, 23, **4**, 522-542 (2018).
11. Bengtsen, S., Getting personal - what does it mean? A critical discussion of the personal dimension of thesis supervision in higher education. *London Review of Educ*., 9, **1**, 109-118 (2011).
12. Maxwell, T.W. and Smyth, R., Higher degree research supervision: from practice toward theory. *Higher Educ. Research & Develop.*, 30, **2**, 219-231 (2011).
13. de Kleijn, R.A.M., Meijer, P.C., Pilot, A. and Brekelmans, M., The relation between feedback perceptions and the supervisor-student relationship in master's thesis projects. *Teaching in Higher Educ*., 19, **4**, 336-349 (2014).
14. de Kleijn, R.A.M., Mainhard, M.T., Meijer, P.C., Brekelmans, M. and Pilot, A., Master's thesis projects: student perceptions of supervisor feedback. *Assessment and Evaluation in Higher Educ*., 38, **8**, 1012-1026 (2013).
15. Martínez, P.J., Aguilar, F.J. and Ortiz, M., Transitioning from face-to-face to blended and full online learning engineering master's program. *IEEE Trans. on Educ*., 63, **1**, 2-9 (2020).
16. Abeyweera, R., Senanayake, N.S., Jayasuriya, J. and Fransson, T.H., A remote mode high quality international master degree program in environomical pathways for sustainable energy systems (SELECT)-pilot program experiences during first year of studies. *Proc. 2018 IEEE Global Engng. Educ. Conf.* (2018).
17. Ahlin, K. and Mozelius, P., Redesign and evaluation of a technology enhanced learning environment for thesis supervision. *ICERI2017 Proc.*, 1, November, 636-643 (2017).
18. Bonaccorso, G., *Machine Learning Algorithms*. Packt (2017).
19. Guleria. P. and Sood, M., Predictive data modeling: educational data classification and comparative analysis of classifiers using Python. *Proc. 2018 Fifth Inter. Conf. on Parallel, Distributed and Grid Computing* (2018).
20. Ciolacu, M., Tehrani, A.F., Binder, L. and Svasta, P.M., Education 4.0 - artificial intelligence assisted higher education: early recognition system with machine learning to support students' success. *Proc. IEEE 24th Inter. Symp. for Design and Technol. in Electronic Packaging* (2018).

21. Zaghloul, A.-R.M. and Saad, A., A unified integrated teaching-learning modular approach to education: application to computer engineering education and to machine learning. *Proc. 32nd Annual Frontiers in Education* (2002).
22. Kondo, N., Okubo, M. and Hatanaka, T., Early detection of at-risk students using machine learning based on LMS log data. *Proc. 6th IIAI Inter. Cong. on Advanced Applied Informatics* (2017).
23. Samin, H. and Azim, T., Knowledge based recommender system for academia using machine learning: a case study on higher education landscape of Pakistan. *IEEE Access*, 7, 67081-67093 (2019).
24. Barnard, H., The Consequences of Improperly Selected Instruments in Hydrometallurgycal Process Plants and the Application of an Automated Decision-Tree for Improved Selection. Engineering Institute of Technology (2019).

BIOGRAPHIES

Yuanyuan Fan received her BSc and MEng degrees from North China Electric Power University in electrical engineering in 2010 and 2013, respectively. She received her PhD degree in electrical engineering from Curtin University, Perth, Australia, in 2017. From 2018 until now, she has been a course coordinator and lecturer at the Engineering Institute of Technology, Perth, Australia. She holds certificates on Python and machine learning. Her research interests include machine learning in education, machine learning in renewable energy and smart grids.

Ana Evangelista received her MSc degree in civil engineering from the Federal University of Rio de Janeiro, in 1996, and her PhD degree in civil engineering from the same university, in 2002. Her PhD research was mostly concentrated on non-destructive tests to evaluate concrete structures. In 1997, she started her academic career coordinating and teaching units in the School of Civil Engineering at the Federal University of Rio de Janeiro, Brazil. From 2016 to 2019, she worked as a visiting research fellow in the area of recycled concrete at Western Sydney University, School of Computing, Engineering and Mathematics. She has published book chapters, 14 referred journal articles and 15 referred conference articles. Currently, she is a lecturer and work integrated learning coordinator at the Engineering Institute of Technology, Perth, Australia.

Hadi Harb received his MEng degree in electrical-electronic engineering from the Lebanese University in 2000. He received his MSc from the Institut National des Sciences Appliquées INSA Lyon, France, in 2001, and his PhD from the Ecole Centrale de Lyon, France in 2004, both in computer science. From 2004 to 2006, he worked in Centrale Lyon Innovation SA as a research engineer. During his PhD and his research engineer work period, he obtained two patents and published 17 articles in refereed international scientific journals and conference proceedings in the area of machine learning application to audio signal analysis. From 2006 to 2015, he founded and managed Ghanni, a company specialised in multimedia content recommendation and identification using artificial intelligence. Several European radio stations and Web sites licensed Ghanni’s music recommendation technology. In 2015, he restructured Ghanni to transform it into a consultancy company in the domain of artificial intelligence, where he acts as the principal consultant. He participated as a consultant in different projects in the domains of natural language processing, personal assistance and audio classification. His current interests are in the use of artificial intelligence techniques to solve industrial problems.