Data Collection for Career Path Prediction Based on Analysing Body of Knowledge of Computer Science Degrees

Ahmad F. Subahi*
Department of Computer Science, University Collage of Al-Jamoum, Umm Al-Qura University, Makkah, Saudi Arabia.

* Corresponding author. Tel.: +966550055856; email: AFSubahi@uqu.edu.sa
Manuscript submitted August 23, 2018; accepted October 21, 2018.
doi: 10.17706/jsw.13.10.533-546

Abstract: Measuring and analysing student performance in higher education are considered essential tasks for improving the quality of degree programs and their graduates. This work investigates a new artificial neural network (ANN) approach for career path prediction (CPP) based on the analysing computer science's body of knowledge (BoK) in degree programs. It proposes a proof-of-concept of a data collection strategy to build the required CPP dataset for a promising data-driven system. An initial design, for validating purpose, of a single-layer ANN is introduced, trained, tested and applied to real-world graduate records to classy them into groups or most appropriate career path for each. The results of the applied experiment show the capability of the proposed CPP approach to classify real-world student records into groups.

Key words: data-driven system, artificial neural network, proof_of_concept, career path prediction dataset

1. Introduction

With the rapid revolution in business innovation and technology, the demand of having skilful IT professionals, who seek successful careers, in various fields of informatics and computing has increased in industry. In general, Computing degrees, such as Computer Science (CS), Software Engineering (SE), Information Technology (IT), Information Systems (IS) and Management of Information Systems (MIS), prepare future professionals to get ready for the job market and to remain competent in industry. Graduates from these degrees are required to have a wide range of constantly updated technical (hard) skills and knowledge in various computing and IT fields. Not only this, but non-technical (soft) skills, business knowledge sets and disciplines are additionally required due to the massive IT impact on business processes and operations [1].

Several model curricula are introduced over the decade and revised many times, such as the Information System model curricula IS2010 [2], Software Engineering SE2014 [3], and Computer Science one CS2013[4], to formulate systematically the critical skills and knowledge required for the computing-related graduates. Besides, many recent research has investigated the mapping between what is taught in these different informatics and computing curricula and which employment skills should be included there [5][6].

In respect to the dynamic nature of the IT and computing field, numerous related studies concluded that future graduates have not adequately prepared, during their university degree, for the job market [7]. Today's IT environment is speedily changing with great shifting in business and industry patterns. This is very challenging for new graduates to obtain necessary technical skills that enable them to perform well in their jobs [8].

Choosing the suitable first job in any computing-related career is one of the most critical decisions should a graduate make. According to the recent growth in the career paths and job opportunities in computing-related domains, making this choice have become tough for the students. The productivity of the graduates in their first job will be affected negatively and reduced because of selecting a wrong job that fits to their skills and knowledge. From that, it is essential to have such an indication that might help graduates to evaluate their skills achieved after completing their computing-related program to encourage them choosing the most appropriate first jobs.

In this work, we investigate the possibility of using artificial neural network (ANN) in predicting the most appropriate career path of graduates from computing-related programs, based on analysing the body of knowledge (BoK) of computer science curricula classified in [4]. The goal of the promising approach is achieving an indication to essential skills that graduates gained after completing their computing-related program. This research has crucial impact on curriculum design, including programs and general education policies. It helps decision makers to find weak areas in their program that might need to be improved to prepare student for the market demands.

To achieve this goal, we devote our work in this paper to produce a new proof-of-concept (PoC) that reflects a formal data collection procedure for obtaining a Career Path Prediction (CPP) dataset. The designed PoC is capable to cover all tasks required in the data pre-processing phase, such as storing new curriculum information, annotating modules with one or more associated KAs including its weights in every module, entering students' total marks from academics, and calculating KA weights for each module based on the stored information of the curriculum in the database.

Furthermore, an initial design of a classification system using artificial neural network (ANN) is presented and discussed, briefly, in this paper as a future work solution to this computational problem (classifying graduates based on their gained computing-related skills). As this intelligent computational side of the system cannot work efficiently without having sufficient amount of data in the CPP dataset to be used in training, validating and testing the promising data-driven system.

The rest of this paper is organised as follows: section 2 represents various related works including other AI and data mining techniques that were adopted in evaluating students' performance approaches and similar contexts. Section 3 highlights different sources of information that support the process of creating the CPP dataset. It includes the data collection and preparation process regarding with student records, course syllabi and recruitment websites. It also includes the algorithm of calculating the weight of each knowledge area in modules in the curriculum. Section 4 describes the overall design of the proposed proof-of-concept (PoC) including design decisions regarding data modelling and HCI design. Section 5 discusses the theoretical foundation behinds the artificial neural network (ANN) technique, including the design choice adopted in this work. Section 6 presents and discusses the detail of our experiment and results.

2. Background and Related Work

In the recent years, tracking student performance and grade in degree programs has been considered in many research investigations, such as [9] and [10]. Three learning features are extracted and analysed, using a feature selection and decision tree technique, from Moodle logs during courses to be used for future grade prediction [9]. On the other hand, a method called course relevance discovery is developed to predict the future student performance using current and past performance of students. Relevant courses are determined using the prerequisite dependencies of courses in a degree program. This method is based on the latent factor model and uses the probabilistic matrix factorization algorithm to perform course clustering [10].

Furthermore, data mining techniques, such as the approach presented in [11], are applied to educational domain and used for solving similar problems. As it is critical to evaluate student performance and investigate the factors that affect their achievements to enhance the quality of education system. Four different classifiers, namely, J48 Decision Tree, Multilayer Perception, Naïve Bayes, and Sequential Minimal Optimization are compared to evaluate the accuracy of classification algorithm in each type [11] using some attributes from a student dataset in Turkey. In their experiment, they came up with an interesting finding demonstrating that the performance of a student is affected by some taught modules that are studied with 87% accuracy.

Moreover, the NLP semantic analysis technique is adopted in [12] and applied to students' reviews for their teacher. This type of data is considered as input for the constructed database for the need of their work. The proposed system can help instructors (teachers) to predict their performance deficiency sides that need to be improved in the future. The achieved by determining the polarity and features of the teacher based upon the considered feedback and comments that are collected from students.

In addition to this, there are various web applications/sites suggest career paths for students. Many of them are based on personality traits and interests of students to predict the career, without considering their capability to judge whether they can survive and proceed in that career path or not. A learning analytics approach is defined and supported by a designed a tool to be used in educational institutes [13] and [14] to predict learner performance and suggest some resources for them to improve their skills. Different types of LSM data attributes in Sakai institute are considered, such as type, date and number of times resource accessed, and student grade on assignment, discussion posts, quizzes and final grade [13]. Similarly, an automated system is developed to evaluate the personality traits of learners [14]. Its classifier, with the c4.5 algorithm, achieved the highest accuracy of 86% using 12 attributes on a students' dataset with 200 records. gives choices to the students for choosing career based upon their skills.

2.1. Student Performance Evaluation

When start thinking about the evaluation of student knowledge and skills in modern higher educational institutions, the first remarkable example, in computer science and engineering domain, springs to mind is the task of evaluating student learning outcomes in the requirements of Accreditation Board for Engineering and Technology (ABET) [15]. ABET has been applying, early, to computer science programs in various U.S.A universities and colleges since the 80s. It gains a significant international recognition and interest over the time [16]. Nowadays, there are about 3,800 accredited programs around the globe, providing approximately 85,000 graduates from ABET-accredited programs every year [15].

ABET is considered in the background of this work because it has, within its overall processes, a core task for assessing and evaluating student learning outcomes at the courses level. There are eleven student learning outcomes a to k related to computer science program that is established to evaluate the students' knowledge and skills, also to inspire the development of student outcomes for each engineering or technology program [15], [17] and [18].

Rubrics are designed for measuring and scoring the student performance. Then a percentage that represents the level of students who are achieving the learning objectives is calculated by course instructors [18]. There are two methods of learning outcomes assessment, namely, indirect and direct. The implementation of the assessment is left as an independent practice applied by each institution, such as [19] and [20]. In the indirect method, the process assesses opinions or thoughts about student knowledge or skills, based on their experience in courses, via surveys and interviews [20] and [18].

Besides, the outcomes are also assessed (direct) through measurable tools in coursework items, such as exam questions and project reports or presentations in each course [17] and [18]. Thus, the overall curriculum can be assessed is based on assessment of all courses in the curriculum, which is conducted by

the instructors and students. Like the indirect method of assessment, the application of the assessment is left as an independent practice performed by each institution. For instance, a course assessment tool is developed, via some enhanced forms, to lead to an effective analysis of assessed course and ABET educational outcomes and objectives [17]. Alternatively, the assessment plan is applied in [19] using some Key Performance Indicators (KPIs) that are designed and tailored to their target program. Each attribute of learning outcomes is parsed into one or more KPIs to be measured.

According to [21], following the common method of grading practical assignments in computer science courses, which is based on the similarity of student answers to the model answers provided by their instructor, leads to ignore the creativity and originality in the student solutions. To tackle this issue, a general programming rubric is developed to be used in grading across all programming courses to enhance more flexibility for grading critical and creative solutions.

The process of development the programming rubric is started with identifying the learning outcomes with respect to the nature of the program and the course. All programming modules are selected involving different programming languages. The learning outcome at program level and the course level are compared and the assessment types, including assignment, practical lab work, group project and final examination. Moreover, percentage of each type is determined across the all courses in the curriculum. The rubric, then, formulated by determining some dimensions, with a relevant scale, that is used for judging the student work. This designed rubric is also evaluated and tested by implementing it during the academic year 2015/2016 in three programming courses, offered by the Faculty of Computer Science and Information Technology (FCSIT) at Universiti Tun Hussein Onn Malaysia (UTHM), showing a very good reliability degree [21].

3. Information to Support Career Path Prediction Dataset

Before going into further detail about data pre-processing strategy, it is worth mentioning that the computer science Body of Knowledge (BoK) is adopted from the most recent review of computer science curricula CS2013[4] to extract the related Knowledge Areas (KAs). The BoK is categorised into eighteen KAs, illustrated in Table 1. These KAs are organised into a set of modules, distributed across the overall curriculum, in which each module contributes to one or more KA, represented via its covered topics. As it is classified by [4], each computing topic taught in any computer science module must serve, somehow, one of these KAs.

Table 1. Computer Science Body of Knowledge

Code	Knowledge Area	Code	Knowledge Area
AL	Algorithms & Complexity	HCI	Human-Computer Interaction
AR	Architecture & Organisation	IAS	Information Assurance & Security
CN	Computational Science	IM	Information Management
DS	Discrete Structures	IS	Intelligent Systems
GV	Graphics & Visualisation	NC	Networking & Communication
OS	Operating Systems	PBD	Platform-based Development
SE	Software Engineering	PD	Parallel & Distributed Computing
PL	Programming Languages	SDF	Software Development Fundamentals
SF	Systems Fundamentals	SP	Social Issues & Professional Practice

The overall data pre-processing strategy can be summaries in the following steps:

- Selecting a computer science program and determine knowledge areas of its modules.
- Implementing the KA weight strategy presented in section 3.2.
- Developing and testing a suitable data model and GUI subsystem for data entry.
- Enabling academics/program coordinators to create a new course plan their selected CS-related program and enter the KA detail for its modules.

- Enabling academics to enter students marks in every assessment item for each module.

Furthermore, the overall data, used in the proposed PoC design is taken from three main sources, namely, academic records of graduate students and course syllabi and recruitment websites. The following subsections discuss the processes of data collection and preparation for each source of information.

3.1. Student Records and Course Syllabi Data

To fulfil the goal of this investigation, course plans (syllabi) of all modules, from the chosen computer science programme1 (academic year 2017-2018), are collected and reviewed. Each syllabus consists of several topics covered throughout the semester. We first held all required syllabi and reviewed their included topics. Then, the percentage of each KA in every module is extracted, by annotating each topic with an associated KA, in similar way as [4].

Besides, academic records of graduates are considered the data source. Each record contains overall marks, GPAs and grades for each taught module, which is unprocessed format. Past exam papers and coursework assessment items, in every module are then collected, from the department archive and the ABET documents of the modules. This step is essential to determine and distribute the percentage of each KA assessed in every item.

After collecting the academic records, the date pre-processing and preparation is achieved through analysing the student performance in each computing-related KA appears in every module in the curriculum and calculating the accumulated percentage for each KA. However, In the produced PoC, thirteen attributes are only considered to be used as an indication to the most suited computing-related career (selected) for the graduates. The following subsection exemplifies the theoretical side of calculation the weight of knowledge areas process.

3.2. The Weight of Knowledge Area

According to the literature covered in this work, every knowledge area (KA) is distributed over a variety of courses in the curricula. From that, it is critical to calculate the weight of each KA in every module appears in the curriculum. This can be achieved via the following formula:

Let $\boldsymbol{M}$ is a finite set of modules in the curriculum $\boldsymbol{C}$, Thus m is a module element belong to $\boldsymbol{M}$ ($m \in \boldsymbol{M}$). Let $\boldsymbol{A}$ is a finite set of assessments in a module (e.g. homework, mid-terms, projects and final exam). Thus, a is an assessment item belong to $\boldsymbol{A}$ ($a \in \boldsymbol{A}$). Let $\boldsymbol{KA}$ is a set of knowledge areas covered in an assessment item a in a module m. From that, the weight $\boldsymbol{W}$ of a ka in a module m can be calculated as:

$$\boldsymbol{W}_{m}^{ka} \coloneqq \sum_{a=1}^{a=n} W_{a}^{ka}$$

Where W_{a}^{ka} is the weight of the ka in the assessment item a. It is worth mentioning that the total W is equal 1. For instance, consider a Compiler Construction module, from the BSc curriculum of Colorado State University, that has 4 assessment items as two mid-term exams, a compiler project and a final exam. There are three main KAs contain topics and learning outcomes covered in this module, namely, Programming Languages (PL), Software Engineering (SE) and Algorithms & Complexity (AL). This can be expressed in three formulas as:

$$\boldsymbol{W}_{Compiler\ Construction}^{PL} \coloneqq (W_{mid-term1}^{PL} + W_{mid-term2}^{PL} + W_{project}^{PL} + W_{final-exam}^{PL})$$
$$\boldsymbol{W}_{Compiler\ Construction}^{SE} \coloneqq (W_{mid-term1}^{SE} + W_{mid-term2}^{SE} + W_{project}^{SE} + W_{final-exam}^{SE})$$

[1] BSc of Computer Science in the University Collage of Al Jamoum, Umm Al Qura University, Makkah, Saudi Arabia (JUC CS)

$$\boldsymbol{W}^{AL}_{Compiler\ Construction} \coloneqq (W^{AL}_{mid-term1} + W^{AL}_{mid-term2} + W^{AL}_{project} + W^{AL}_{final-exam})$$

A module coordinator sets the number of marks for each KA in each assessment item. For instance, suppose the full mark of the $mid - term1$ is 20. It means the weight of the $mid - term1$ in the module is 20% out of 100. When two questions that cost 10 marks cover the PL knowledge area, it means that the weight of PL in $mid - term1$ ($W^{PL}_{mid-term1}$) is 50% of the weight of the $mid - term1$, which equals to 10% or 0.1 in overall out of 100.

The rest weights ($W^{SE}_{mid-term1}$) and ($W^{AL}_{mid-term1}$) are calculated similarly. This process is applied to every assessment item in the module, which are $mid - term2, project$ and $final - exam$. The following table (table 2) demonstrates an example of anonymous student marks in all assessment items in the Compiler Construction module.

Table 2. Example of Calculation KA Weights from Student Marks

Assessment Item (out of 100)	PL		SE		AL	
	Std mark	W	Std mark	W	Std mark	W
20% Mid-term1	10	0.1	2	0.04	5	0.06
20% Mid-term2	7	0.08	1	0.04	3	0.08
20% Project	6	0.06	4	0.06	7	0.08
40% Final	12	0.16	10	0.12	11	0.12
Total	35	0.4	17	0.26	26	0.34

When applying the above formulas to the student data presented in Table 2 above, the weight of each KA is calculated.

$$\boldsymbol{W}^{PL}_{Compiler\ Construction} \coloneqq 0.1 + 0.08 + 0.06 + 0.16 \coloneqq 0.4$$
$$\boldsymbol{W}^{SE}_{Compiler\ Construction} \coloneqq 0.04 + 0.04 + 0.06 + 0.12 \coloneqq 0.26$$
$$\boldsymbol{W}^{AL}_{Compiler\ Construction} \coloneqq 0.06 + 0.08 + 0.08 + 0.12 \coloneqq 0.34$$

The percentage of student performance in the PL knowledge area in the Compiler Construction module $P^{PL}_{compiler\ consturction}$ can be calculated via the following formula:

$$\boldsymbol{P}^{PL}_{compiler\ consturction} = \frac{total\ PL\ student\ marks}{\boldsymbol{W}^{PL}_{Compiler\ Construction}}$$

$$\boldsymbol{P}^{PL}_{compiler\ consturction} = \frac{35}{0.4} = 87.5\%$$

In a similar way, the percentage of student performance in the SE and AL knowledge areas in the Compiler Construction module $P^{AL}_{compiler\ consturction}$ and $P^{SE}_{compiler\ consturction}$ can be calculated as:

$$\boldsymbol{P}^{AL}_{compiler\ consturction} = \frac{17}{0.26} = 65.4\%$$

$$\boldsymbol{P}^{SE}_{compiler\ consturction} = \frac{26}{0.34} = 76.5\%$$

To accomplish calculating the performance of the student in a single KA, let's say PL, the above steps, applied to Compiler Construction module, must be applied to the rest of curriculum modules that are studied by the student and cover PL knowledge area. This strategy is applied to all to all KAs and modules in similar way. At the end a list of the complete performance of each KA is achieved for each student.

3.3. Recruitment Websites Data

Some recruitment websites that advertise a variety of jobs such as [22], [23] and [24] are reviewed. Then, some common computing-related positions are selected to form several suggested career paths, such as Database designer, Software Engineer, Web application developer, Networking and Business analyst. Next, the description of each selected job is manually inspected to extract required skills and knowledges of qualified candidates for each job type.

The extracted skills are divided into major and specific skills. Common skills that appears in the job descriptions of a specific position across all reviewed websites are considered major skills. On the other hand, all job-specific skills, for the same job, across those sites are considered specific skills. At the end of these reviews and comparisons, each selected computing-related job must have at least one major skill and many specific (preferable) skills. The following UML Activity diagram (Figure 1) demonstrates the whole process that are discussed in the following subsections.

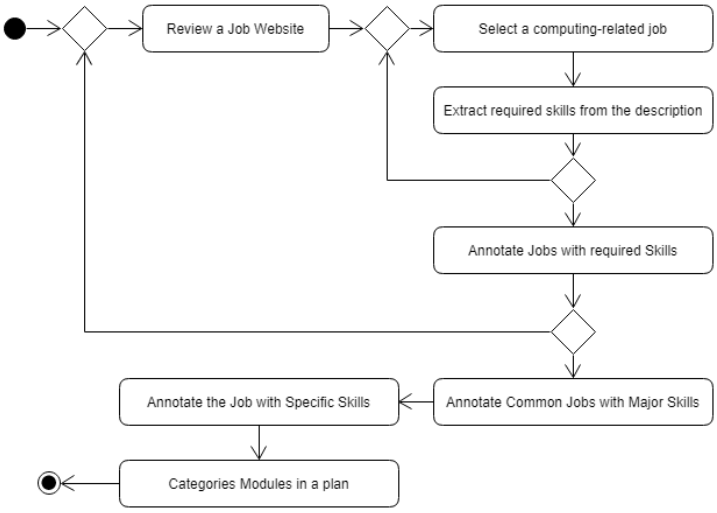

Fig. 1. The whole process of preparing recruitment websites data.

The previously determined KAs are used, here, for annotating the major and specific skills mentioned above. Thus, a list of jobs with several required (core) and optional (preferable) KAs for each position is achieved. In this PoC implementation, only core KAs and two kinds of computing-related job positions are considered for the validity of the proposed idea, namely, Business Analyst and Networking. The following table (Table 3) summarise extracted jobs with their associated core skills:

Table 3. The job-skill mapping summary

Job Position	Required Skills (KAs)
Networking	NC, OS, ISA, PD, AR
Business Analyst	IM, HCI, SP, SE
Database Administrator	IM, ISA
System Developer	AL, AR, SF, SDF, OS, PD, NC
Data Analyst	IS, CN

4. Proof of Concept Design

The promising PoC should have some GUI screens that achieve an effective human-computer interaction in data entry. Moreover, it must store huge amount of data in a database with a good query performance. As we are aiming at presenting a heterogenous framework for the prediction system at the end, it is recommended using Python, specially the PyQt and NumPy packages. This section highlights some of the

important design choices that are be adopted in design our solution from two perspectives, data modelling and GUI design.

4.1. Data Modelling Strategy

The proposed PoC can be categorised as an information system, therefore data modelling is one of the most critical activity conducted during the development process. The following UML Class diagram (Figure 2) represents the data model (schema) of the overall system. It can be noticed that is a long chain of associations to represent the relationships among several entity types. This means that these entities (classes) are functionally connected and the database performance is affected. The cost in the performance occurs when dealing with queries that require many joins. De-normalisation in data modelling strategy is one possible method can be adopted to increase the overall performance of database.

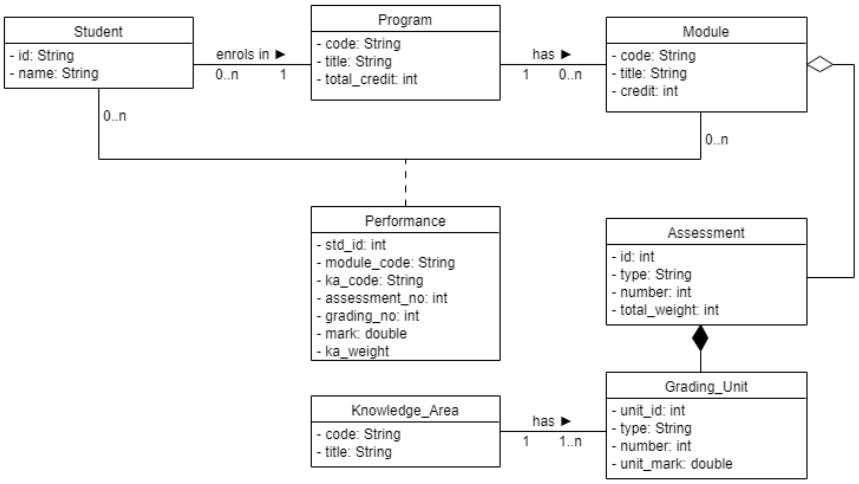

Fig. 2. The data model of the proposed PoC.

However, as a design decision to be adopted in this work, graph database can be chosen for data modelling to speed up the queries and increase the overall performance. This work suggests Neo4j database application [25], one of the well-known open-source graph database management system (GDBMS). Neo4j stated clearly some basic step for converting relational schema into graph database one. The above relational schema is used as an inspiration of the construction process of the graph data model in Neo4j.

As a graph database seeks only records that are connected to other records without considering the number of records in the table. It is also beneficial in graph databases that data schema can be either predefined or constructed/modified on the fly. As the final system is still evolving, having a flexible data model to application features and business logics change, which can be created on the fly, is preferable design choice that increases the speed of the development process. Consequently, two main scenarios are considered for designing an initial graph data model that might be modified during progress of the development stages. The first main scenario of the system can be expressed using natural language as:

*A **module**, in the academic **program**, has different **assessment items**,*
*each item consists of one or more **grading unit(s)**,*
*each unit is associated with one **knowledge area**.*

Additionally, the second core scenario can be written as:

*Every **student**, enrolled in the **program**,*
*studies many **modules** and performs in the module assessments.*

The following figure (Fig. 3) illustrates the representation of the two core scenarios using a graph data model.

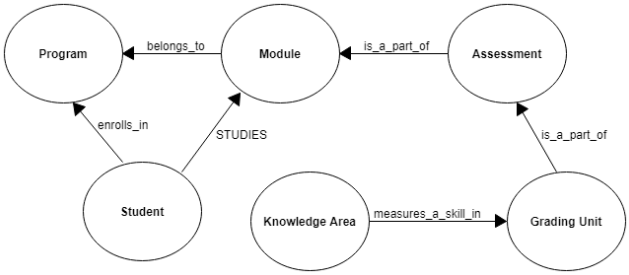

Fig. 3. The graph representation data model.

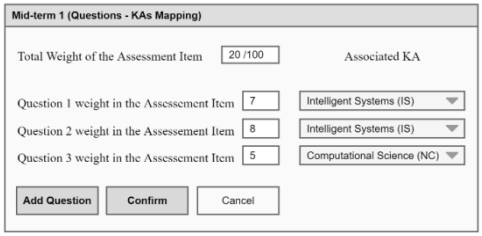

Fig. 4. Setting KA weights for an assessment.

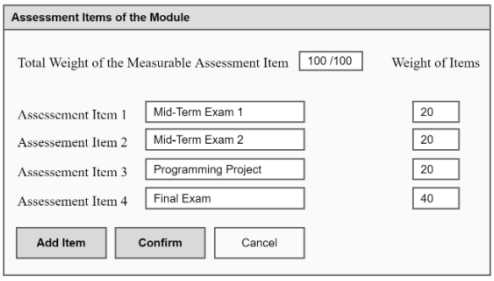

Fig. 5. Setting a new assessment item.

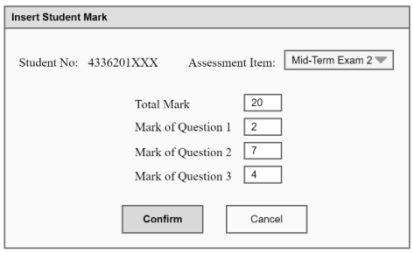

Fig. 6. Stetting student's marks in an assessment item.

4.2. Graphical User Interface Design Strategy

In this section, appropriate mock-ups that represent system GUIs are designed to facilitate the process of entering into the system the KA information and weights for all modules, as well as students' performance (marks) for each one. As we are aware that academics in the domain don't measure directly the computing-skills of students using assessment items for the courses. They, instead, measure their outcomes as a part of the academic accreditation process. In the accreditation, some exam questions are selected by the academics to measure each outcome separately.

In our approach, we are trying to adopt this strategy and allow academics to annotate each question in every exam by one KA. The following mock-ups (Fig. 4 and 5) illustrate how a program coordinator or an academic can allocate assessment items to a module, as well as how they can determine the percentage of KA in each item.

Besides, an academic can enter marks of a student in every assessment item designed for each module using the screen represented in Fig. 6 below:

As we mentioned earlier in this paper, to achieve the automated prediction by adding an intelligent value to the proposed PoC, it would be beneficial to use the widely adopted artificial neural networks (ANN) technique to build an AI classification system. The promising ANN model uses the accumulated percentage of each knowledge area rather than the obvious GPA or marks in modules, for each graduate record. It is worth mentioning that these accumulated percentages are calculated via the proposed PoC tool. The following section highlights the theoretical foundation behind the suggested design of an ANN to be enhanced and implemented in the future work.

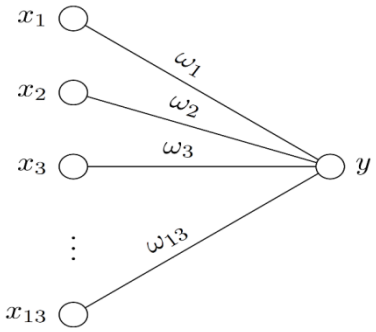

Fig. 7. The designed single layer ANN for CPP.

5. Theory of Artificial Neural Networks

This section highlights some future work aspects regarding a well-known machine learning technique, neural networks, that can be considered in developing the second phase of the system after data pre-processing, including training, validating and testing. Artificial Neural Network (ANN) is a computer program designed to simulate the way in which the human brain processes information. It is a highly adopted machine learning technique for approximation in various range of application domains, such as [26 - 28]. The most appropriate type of the ANN is determined based on the nature of the problem to be solved. However, a forward propagation neural network with the backpropagation algorithm is adopted in most applications.

In the forward ANN, the information moves in only the forward direction without forming any cycle or loop, from the input nodes to the output ones through some hidden nodes. To reduce the error, which is computed using a cost function, a backpropagation algorithm, the gradient descent algorithm, is then applied backwards to adjust the weights based on the computed error. Thus, the provided result in the next time becomes more accurate.

As this work is aiming at presenting a proof-of-concept of the prediction approach, a simple forward propagation ANN with a backpropagation algorithm is considered, created and trained using a very limited number of records, only to examine the validity of our classification strategy for identifying the most appropriate career paths of graduates based on their performance.

As multi-layer ANN is very popular in the domain of classification, we designed the simplest form of the multi-layer ANN, which is called a single-layer ANN that consists of some neurons and connections distributed into two main layers: input and output layer, without any hidden layer. The model can be formally defined as:

Giving $(\mathcal{N}, \mathcal{V}, w)$ where $\mathcal{N}, \mathcal{V}$ are two sets, a set for neurons and a set for connections between neurons respectively, and a function w to define the weight on each connection between each couple of neurons. Let $i, j \in \mathcal{N}$, therefore $w(i,j)$ is the weight on the connection between the two neurons i and j. This can be written as $w_{i,j}$.

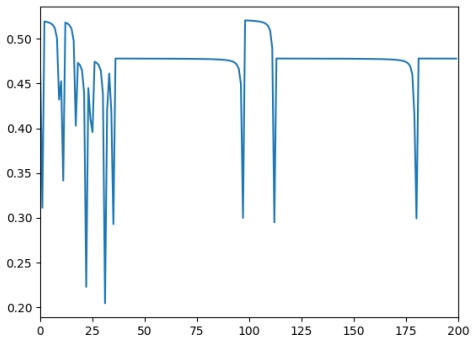

Fig. 8. Plot of the MSE versus the number of epochs (200) for training data for the designed ANN.

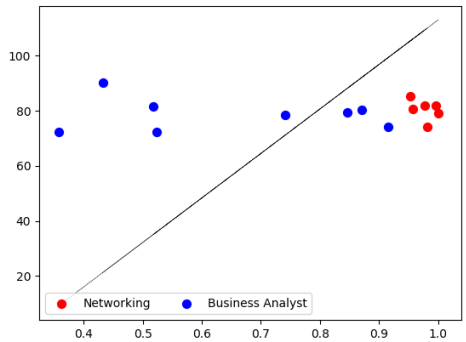

Fig. 9. Regression plot of trained ANN showing the relationship between targets (Y) and the output (X) with epochs = 200.

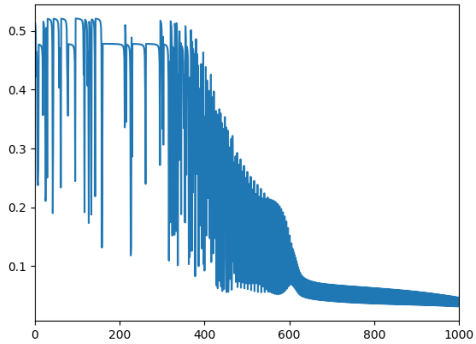

Fig. 10. Plot of the MSE versus the number of epochs (1000) for training data for the designed ANN.

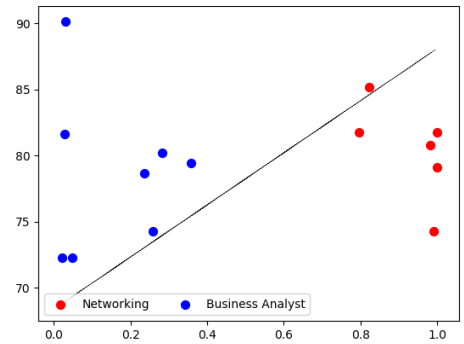

Fig. 11. Regression plot of trained ANN showing the relationship between targets (Y) and the output (X) with epochs = 1000.

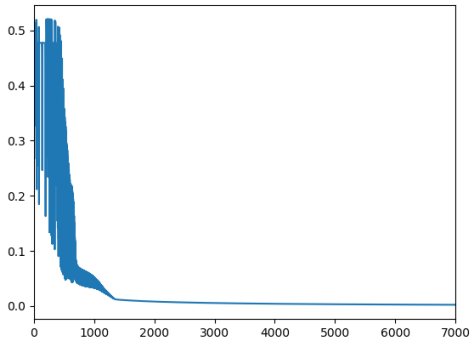

Fig. 12. Plot of the MSE versus the number of epochs (7000) for training data for the designed ANN.

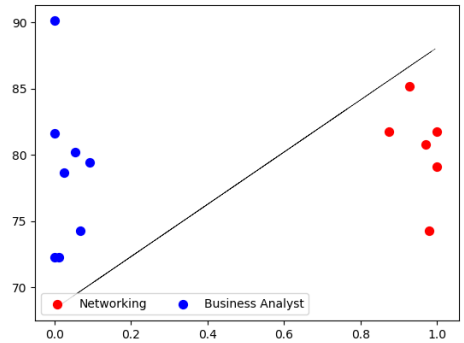

Fig. 13. Regression plot of trained ANN showing the relationship between targets (Y) and the output (X) with epochs = 7000.

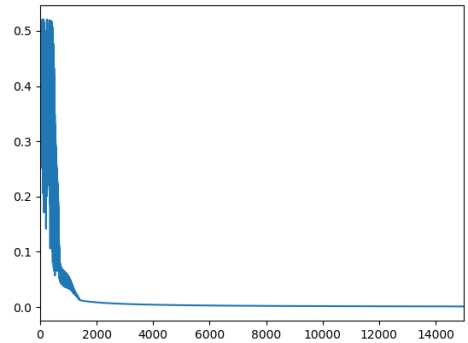

Fig. 14. Plot of the MSE versus the number of epochs (15000) for training data for the designed ANN.

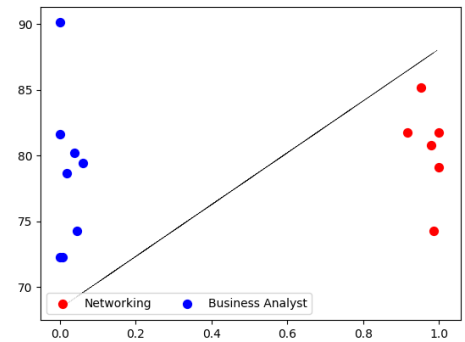

Fig. 15. Regression plot of trained ANN showing the relationship between targets (Y) and the output (X) with epochs = 15000

The input layer consists of 13 input neurons for a single graduate record. Each input is multiplied by a random weight (in the forward propagation training), which is adjusted later during the backpropagation training. The sum of the thirteen inputs multiplied by their weights are transferred to the next layer through a mathematical function. As we implement our ANN using Python NumPy package, the sigmoid function is used as an activation function between the ANN layers.

As a design decision of the proposed ANN, we used a single node in the output layer with a binary classifier (0, 1) to be computed using sigmoid function at the current demand of the PoC development as it is demonstrated in Fig. 7. Therefore, the proposed ANN model can classify career paths into only two categories after applying different numbers of epochs in model training process.

6. Experiment Results and Dscussion

The main goal of the experiment is to examine the validity of applying the idea of classifying computer science graduates based on the distributed KA on various modules in their program to find the most appropriate computing-related career path for them.

Here, automated prediction of the most appropriate computing-related career path for graduates has been implemented using the ANN technique. Marks and grades of 23 graduates, from JUC CS program in the academic year 2017/2018 graduates, are considered and used for training the designed ANN. The graduate records are selected randomly to avoid any bias in the training data, as it summarised in Table 4. It is worth mentioning that we are aware of the used dataset size might lead to cause an overfitting problem. However, in the proposed PoC we aim at achieving the classification of records in the test data set.

As it can be seen in the following table, 13 attributes are collected in an excel document. The initial preprocessing task is manually performed by completing missing cells in data by standard values. Any inconsistence values and irrelevant attributes are manually removed to enhance the classifier quality.

Table 4. Snapshot of CPP Dataset Including the Thirteen Parameters (KAs) as the Input of the Designed ANN

NC	OS	ISA	PD	AR	IM	HCI	SP	SE	PL	AL	IS	CN	Output
80	89.17	72	63	79.45	87.36	94.71	84.44	85	86.67	75.97	82	70	0
90	71.67	84.66	86.67	75.89	65.81	62	65	78.61	74.33	75.55	60	63	1
75.5	79.41	88.11	82.44	78.89	81.63	85.7	79.85	82	81.43	83.7	78.54	71	0
90	70	90	90	55.45	54.85	76.91	71.25	57.78	60	52.83	90	90	1
60	70	60	92	55.45	90.85	71.25	76.91	87.6	60	71.25	63	80	0
66.14	70	56.75	60	58.95	69.32	85	64.44	70.28	68.33	58.5	70.25	65	0
90	71.67	84.66	86.67	75.89	75.81	82	75	78.61	64.33	75.55	80	70	1
88.67	66.5	81.5	66.75	71.55	75.5	61.5	88.54	80.42	82.5	67.26	71.34	88	1
81	57.68	80.1	76.4	63.13	66.21	73.93	80.33	76.54	55.4	58.22	63.5	52	0
79.27	76.67	73.72	75	76.67	80.54	82.5	77.92	76.67	61	68.61	65	75	0
65.75	73.72	77.54	70.11	86.73	82.25	77.54	68.6	86.12	74.66	54.79	86.96	75	0
77.88	82.5	85.22	68.77	69.74	79.59	61.48	83.12	77.53	87.33	63.96	72.34	85	1
66.14	70	56.75	60	58.95	79.32	75	84.44	70.28	68.33	58.5	70.25	65	0
74.46	59.63	79.51	68.5	65.14	82.74	72.84	66.74	68.5	71.12	72.25	76.11	80	0
65.75	93.72	77.54	80.11	86.73	82.25	70.54	78.6	76.12	74.66	54.79	86.96	75	1
82	88.53	91.37	78.11	86	85.68	90.21	73.25	65.17	70.89	75.97	68.9	62.47	1
69.27	76.67	73.72	75	76.67	64.54	62.5	77.92	66.67	61	68.61	65	75	1
70.46	85.62	87.12	80.5	59.6	90.75	87.54	73.52	79.5	67.14	73.54	69.5	78	0
86.5	90	64.56	82.36	66.25	68.5	64.56	69.77	73.15	69.79	81.89	74.59	75	1
65.17	68.9	81.54	78.11	86	85.68	78.53	83.25	91.37	83.75	75.97	70.89	62.47	0

The implemented ANN includes two layers, one input layer with 13 neurons and one output layer with only one neuron. In the experiment, we investigate the effect of using for different numbers of epoch (20 0,

1000, 7000 and 15000) in the network training and measure its performance using the value of the mean squared error (MSE). As it can be seen in the following figures (7, 9, 11 and 13) that the designed ANN achieved its best performance after 1000 epochs and it remains the same. Regarding the classification result, as it is demonstrated in figures (8, 10, 12 and 14) below that test data is classified, clearly, into two categories, after using over 1000 epochs, as it is expected.

7. Conclusion and Future Works

In this work, the possibility of using artificial neural networks for career path prediction (CPP) was investigated via introducing a data collection strategy to build the CPP dataset. The overall design of a suggested proof-of-concept, that implement the data collection procedure, is introduced using python programming language and Neo4j graph database technology. Additionally, the feedforward backpropagation single-layer ANN was used in this investigation to evaluate the validity of the proposed idea. An experiment that aims at obtaining the highest possible classification results of testing set samples is designed.

The experiment results support our claim that the ANN can be utilised for career path prediction when it applies to classify students into two target categories using the determined KAs covered in their degree. Based on that, it can be also claimed that an enhanced version of the proposed PoC and ANN can cover more career paths in the prediction. The result of this initial investigation should be considered and improved in the future work after enlarging the size of the CPP dataset and improve the structure of the ANN and its training algorithm.

References

[1] Amadi, E. U. (2012). Information systems program and business needs: Case study of a Midwestern University (Doctoral dissertation, Capella University).

[2] Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker Jr, J. F., Sipior, J. C., & de Vreede, G. J. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, *26(1)*, 18.

[3] Ardis, M., Budgen, D., Hislop, G. W., Offutt, J., Sebern, M., & Visser, W. (2015). SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering. *Computer*, *48(11)*, 106-109.

[4] Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science. ACM New York, NY, USA.

[5] Stevens, D., Totaro, M., & Zhu, Z. (2011). Assessing IT critical skills and revising the MIS curriculum. *Journal of Computer Information Systems*, *51(3)*, 85-95.

[6] Pratt, J. A., Keys, A., & Wirkus, T. (2014). Preparing information systems graduates for a complex society: Aligning IS curricula with liberal education learning outcomes. *Journal of Information Systems Education*, *25(1)*, 35.

[7] Weber, R. (2004). Some implications of the year-2000 era, dot-com era, and offshoring for information systems pedagogy. *MIS Quarterly*, *28(2)*, iii-xi.

[8] Fang, X., Lee, S., & Koh, S. (2005). Transition of knowledge/skills requirement for entry-level IS professionals: An exploratory study based on recruiters' perception. *Journal of Computer Information Systems*, *46(1)*, 58-70.

[9] Figueira, A. (2016, July). Predicting grades by principal component analysis: A data mining ap proach to learning analyics. *Proceedings of the 2016 IEEE 16th International Conference on In Advanced Learning* Technologies (pp. 465-467).

[10] Xu, J., Moon, K. H., & Van, D. S. M. (2017). A machine learning approach for tracking and predicting student performance in degree programs. *IEEE Journal of Selected Topics in Signal Processing*, *11(5)*, 742-753.
[11] Ahmed, A. M., Rizaner, A., & Ulusoy, A. H. (2016). Using data mining to predict instructor performance. *Procedia Computer Science*, *102*, 137-142.
[12] Agaoglu, M. (2016). Predicting instructor performance using data mining techniques in higher education. *IEEE Access*, *4*, 2379-2387.
[13] Dietz-Uhler, B., & Hurn, J. E. (2013). Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of Interactive Online Learning*, *12(1)*, 17-26.
[14] Katore, L. S., Ratnaparkhi, B. S., & Umale, J. S. (2015, April). Novel professional career prediction and recommendation method for individual through analytics on personal traits using C4. 5 algorithm. *Proceedings of the 2015 Global Conference on Communication Technologies* (pp. 503-506).
[15] About ABET. (2018, May 28). Retrieved from http://www.abet.org/about-abet/
[16] Lu, C., Lidtke, D. K., Dierbach, C., & Meiselwitz, G. (2002). Recent experience with the computer science accreditation process. *The International Federation for Information Processing*.
[17] Alhassan, M., & Ashur, S. (2013). Developing course assessment tool to measure the degree of achieving course learning outcomes. *Journal of Systemics, Cybernetics and Informatics*, *11(6)*, 71-74.
[18] Abdallah, M. (2015, June). Student outcomes assessment and evaluation for ETAC/ABET. Paper presented at 2015 ASEE Annual Conference & Exposition, Seattle, Washington.
[19] Aly, W. H. F. (2012). Experience towards ABET educational accreditation at CS program in imam University.
[20] Al-Yahya, S. A., & Abdel-Halim, M. A. (2013). A successful experience of ABET accreditation of an electrical engineering program. IEEE Transactions on Education, *56(2)*, 165-173.
[21] Mustapha, A., Samsudin, N. A., Arbaiy, N., Mohammed, R., & Hamid, I. R. (2016). Generic assessment rubrics for computer programming courses. *Turkish Online Journal of Educational Technology-TOJET*, *15(1)*, 53-68.
[22] Find a Job. (2018, June 5). Retrieved from http://www.jobs.ac.uk
[23] IT Jobs. (2018, June 6). Retrieved from https://www.irishjobs.ie/Jobs/IT
[24] IT Jobs. (2018, June 8). Retrieved from https://www.jobs.ie/it_programming_jobs.aspx
[25] Why graph databases?. (2018, June 22). Retrieved from https://neo4j.com/why-graph-databases/
[26] Kohail, S. N. (2012, March). Using artificial neural network for human age estimation based on facial images. *Proceedings of the 2012 International Conference on Innovations in Information Technology* (pp. 215-219).
[27] Yang, C. P., Hsieh, C. Y., Wang, Y. M., & Hsiao, W. P. (2011, April). Using artificial neural network for reservoir water quality analysis in Taiwan. *Proceedings of the 2011 International Conference on Consumer Electronics, Communications and Networks* (pp. 1883-1886).
[28] Ahmad, A., & Anderson, T. (2014). Global solar radiation prediction using artificial neural network models for New Zealand. Solar 2014 Conference & Expo.

Ahmad F. Subahi is an assistant professor and the head of Department (HoD) at Department of Computer Science, University Collage of Al Jamoum, Umm Al-Qura University, Makkah, Saudi Arabia. He received his PhD from the department of Computer Science, University of Sheffield in 2015. His research is focused on software engineering and computer science. Specific research areas include model-driven engineering, domain-specific modelling languages, code generation and programming languages design.