Education and Information Technologies (2023) 28:3937–3952
https://doi.org/10.1007/s10639-022-11075-8

Identifying digital capabilities in university courses: An automated machine learning approach

Zongwen Fan[1,2] · Raymond Chiong[2]

Received: 28 August 2021 / Accepted: 21 April 2022 / Published online: 6 October 2022

Abstract

Digital capabilities have become increasingly important in this digital age. Within a university setting, digital capability assessment is key to curriculum design and curriculum mapping, given that digital capabilities not only can help students engage and communicate with others but also succeed at work. To the best of our knowledge, however, no previous studies in the relevant literature have reported the assessment of digital capabilities in courses across a university. It is extremely challenging to do so manually, as thousands of courses offered by the university would have to be checked. In this study, we therefore use machine learning classifiers to automatically identify digital capabilities in courses based on real-world university course rubric data. Through text analysis of course rubrics produced by course academics, decision makers can identify the digital capabilities that are formally assessed in university courses. This, in turn, would enable them to design and map curriculums to develop the digital capabilities of staff and students. Comprehensive experimental results reveal that the machine learning models tested in this study can effectively identify digital capabilities. Among the prediction models included in our experiments, the performance of support vector machines was the best, achieving accuracy and F-measure scores of 0.8535 and 0.8338, respectively.

Keywords Digital capability identification · University course rubrics · Machine learning models · Text pre-processing

✉ Raymond Chiong
raymond.chiong@newcastle.edu.au

Zongwen Fan
zongwen.fan@hqu.edu.cn

[1] College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China

[2] School of Information and Physical Sicences, The University of Newcastle,
Callaghan NSW 2308, Australia

1 Introduction

Digital capabilities – recognised as key skills that students must possess to learn and work in an increasingly digital world – have received increasing research attention in recent years (Crosby et al., 2020; Wilson & Slade, 2020). To thrive at university studies, students must be equipped with the skills necessary to use various technologies appropriately and effectively in different spaces, places and situations (Elphick, 2018). Digital capabilities not only can help students to engage and communicate with others in personal life but also to succeed at their workplace later (Krasuska et al., 2020). For example, employers' attention to the digital capabilities of their current and potential employees is rising, since almost every organisation is reliant on such capabilities in transitioning between the various maturity model levels (González-Rojas et al., 2016). The skills required to create documents, presentations and spreadsheets, and to communicate via email and social media, are crucial components of human capital because highly skilled users are better positioned to benefit from using the Internet (Zhong, 2011).

Given the above, it is crucial to include digital capabilities in courses offered by universities. One way to identify digital capabilities in university courses is through the assessment rubrics that course coordinators/academics produce (Pagani et al., 2016). Individual features of assessment rubrics comprise the most direct source of information available to quantify the range of digital capabilities assessed across the student journey (Whetstone & Moulaison-Sandy, 2020). However, manual methods currently in place to quantify students' attainment of digital capabilities, especially when thousands of courses need to be considered, are highly inadequate (Edwards & Fenwick, 2016). Building an automated process for this purpose is thus essential.

In this study, we aim to identify digital capabilities in all courses offered by the University of Newcastle, Australia, using machine learning classifiers by analysing the course rubric data. Digital capabilities considered here include data, information and computer literacy. Based on the classification results, decision makers can design and map curriculums to develop the digital capabilities of both staff and students. By using findings from text analysis of course rubrics produced by course coordinators, decision makers would gain understanding about where and what digital capabilities are formally assessed in the courses.

However, manually processing and labelling all the data to be used for such an analysis would be difficult, since there are about 4,000 courses being offered across the University. In addition, experts would find it time-consuming to label all the samples from all the courses, given the massive data obtained from each of the courses. We therefore apply machine intelligence to analyse rubrics that contain criteria or performance descriptors related to digital capabilities, because using machine learning methods allows us to automatically process and label the data by building prediction models (Balyan et al., 2020). The prediction models built would help identify the digital capabilities in different courses and, in turn, reveal (a) formative tasks that address, but do not formally assess, digital capabilities; (b) gaps in digital capability assessments in programs; and (c) models of good practice in the assessment of digital capabilities across the university. Knowing these details will lead to better designed curriculums for today's digital society and students equipped with key digital skills required for their future career development.

The remainder of this paper is organised as follows. In Section 2, we introduce the machine learning models used in this study in detail. These include two standard single classifiers—Artificial Neural Network (ANN) and Support Vector Machine (SVM); and two ensemble classifiers—Random Forest (RF) and eXtreme Gradient Boosting (XGBoost). Next, we present the experimental setup and report the results in Section 3. Finally, we draw conclusions in Section 4 and highlight some future research directions.

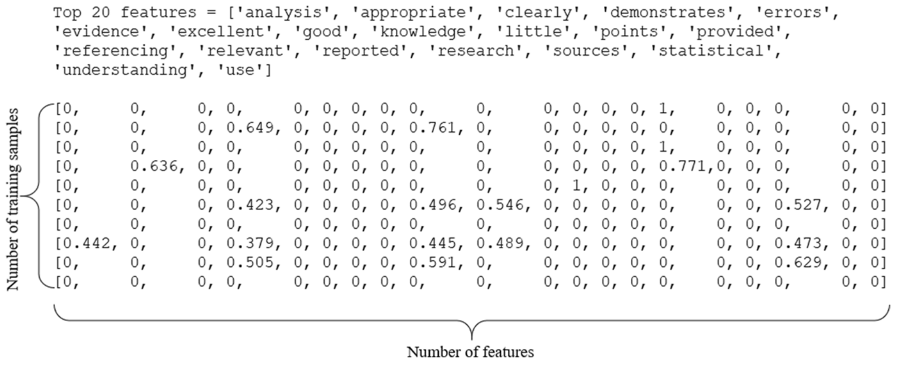
2 Methods

In this section, we first introduce the text pre-processing techniques that we used to pre-process the course rubrics, and then present the four widely used machine learning models. Of note, predicting digital capabilities based on course rubrics is inherently difficult because of the complexity and ambiguity of natural language.

2.1 Text pre-processing

Text pre-processing is an important step in automatically analysing a text dataset. It allows machine learning models to obtain structured information based on text data (Balyan et al., 2020). In this regard, the bag-of-words technique is widely used to encode text data such that it is ready for use by algorithms (Qader et al., 2019). Based on this technique, features can be extracted by mapping from textual data to real-valued vectors. To be more specific, given a text dataset, first, a list of unique words (vocabulary) is prepared from the dataset. Then, one-hot encoding is used to represent each sentence or document as a vector, using the values 1 and 0 to indicate that a word is present in, and absent from, the vocabulary, respectively (Brownlee, 2017). Let us consider the sentence "Collection of data. Are the data used for the model well specified and collected?" – the unique words in this sentence are [collection, of, data, are, the, used, for, model, well, specified, and, collected].

The term frequency-inverse document frequency (TF-IDF) technique, which counts the number of times each word appears in a document, is an effective text representation method (Qaiser & Ali, 2018; Tang & Liao, 2021). Here, TF = (Number of times each unique word appears in a document)/(Number of terms in the document) and IDF = $log(N/n)$, where N is the number of documents and n is the number of documents in which the word has appeared. The TF-IDF value of a term is TF * IDF. An example of TF-IDF data pre-processing can be found in Fig. 1. Note that the figure only shows the top 20 words used as the feature length, but in our study, much larger values have been used (e.g. 500, 800, 1,000, 1,500 and 2,000) to search for the best number of features for the prediction models.



```
Top 20 features = ['analysis', 'appropriate', 'clearly', 'demonstrates', 'errors',
'evidence', 'excellent', 'good', 'knowledge', 'little', 'points', 'provided',
'referencing', 'relevant', 'reported', 'research', 'sources', 'statistical',
'understanding', 'use']

[0,     0,     0, 0,     0, 0, 0, 0, 0,     0,     0, 0, 0, 0, 1,     0, 0, 0,     0, 0]
[0,     0,     0, 0.649, 0, 0, 0, 0, 0.761, 0,     0, 0, 0, 0, 0,     0, 0, 0,     0, 0]
[0,     0,     0, 0,     0, 0, 0, 0, 0,     0,     0, 0, 0, 0, 1,     0, 0, 0,     0, 0]
[0,     0.636, 0, 0,     0, 0, 0, 0, 0,     0,     0, 0, 0, 0, 0.771,0, 0, 0,     0, 0]
[0,     0,     0, 0,     0, 0, 0, 0, 0,     0,     0, 1, 0, 0, 0,     0, 0, 0,     0, 0]
[0,     0,     0, 0.423, 0, 0, 0, 0, 0.496, 0.546, 0, 0, 0, 0, 0,     0, 0, 0.527, 0, 0]
[0,     0,     0, 0,     0, 0, 0, 0, 0,     0,     0, 0, 0, 0, 0,     0, 0, 0,     0, 0]
[0.442, 0,     0, 0.379, 0, 0, 0, 0, 0.445, 0.489, 0, 0, 0, 0, 0,     0, 0, 0.473, 0, 0]
[0,     0,     0, 0.505, 0, 0, 0, 0, 0.591, 0,     0, 0, 0, 0, 0,     0, 0, 0.629, 0, 0]
[0,     0,     0, 0,     0, 0, 0, 0, 0,     0,     0, 0, 0, 0, 0,     0, 0, 0,     0, 0]
```

Fig. 1 An example of TF-IDF processing of feature extraction with the top 20 words used as the feature length

2.2 Machine learning

Machine learning is among the most important areas of artificial intelligence that provides prediction models the ability to automatically learn and improve from experience without explicit programming (Bishop, 2006; Borges et al., 2020). Traditional machine learning classifiers include standard single and ensemble classifiers.

2.2.1 Standard classifiers

ANNs These have been widely used in many practical applications and are able to simulate the way the human brain analyses and processes information (Zhu et al., 2019). Typically, ANNs have three types of layers – the input, the hidden and the output layers. Specifically, for the input layer, the number of neurons is the same as the number of input features, whereas the output layer is the output of the model, usually with only one neuron for binary classification. Neurons in the hidden layer lie in between the input and output layers and are interconnected with both. Those interconnected neurons are able to exchange messages with each other. The ANN is trained based on the tuning of weights between the neuron connections.

SVMs Introduced by Vapnik and Chervonenkis in the early 1960s (Burges, 1998), SVMs are based on statistical learning theory and structural risk minimisation theory (Chiong et al., 2021). It is a supervised machine learning model. Given a dataset with n examples $\{(x_i, y_i)\}_{i=1}^{n}$, the SVM aims to find a hyperplane in a high-dimensional space to separate the samples (Land & Schaffer, 2020). The quadratic programming of SVMs can be expressed as:

$$\begin{aligned} &\arg\min_{w,b,\xi_i} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\ &s.t. \quad \begin{cases} y_i(w^T \phi(x_i) + b) - 1 + \xi_i \geqslant 0, \\ \xi_i \geqslant 0 \end{cases} \end{aligned} \tag{1}$$

where ξ_i is the i th slack variable, C is a penalty parameter, x_i is the i th input vector, y_i is the i th output value, $\phi(x)$ is a mapping function that maps x to the higher dimensional space, w and b are the weight vector and bias of linear function $f(x) = w\phi(x) + b$, respectively, and w^T is the transpose of w.

2.2.2 Ensemble classifiers

RF The RF, a supervised learning algorithm, is an ensemble model of tree predictors (Breiman, 2001) built on bootstrap samples. Its performance in addressing classification as well as regression problems has been promising (Gislason et al., 2006). First, the RF model resamples several training sets based on given data, each set consisting of the same number of samples. Then, it trains decision tree classifiers from the resampled training sets. The random selection of features increases model diversity, which is very helpful in alleviating overfitting issues when aggregating the classifiers for final prediction.

XGBoost This optimised distributed gradient boosting library is designed to be highly efficient, flexible and portable (Chen et al., 2015). It provides parallel tree boosting (also known as gradient boosted decision tree and gradient boosting machine), which solves many data science problems rapidly and accurately. Similarly to the RF, XGBoost is also able to mitigate the overfitting problem (Chen et al., 2015). It applies a more regularised model formalisation using gradient boosting (Friedman, 2002) to improve its performance. In addition, it has useful properties for real-world applications, including the column block for parallel learning, cache-aware access and blocks for out-of-core computation (Carmona et al., 2019).

3 Experiments and results

In this section, we compare the performance of different machine learning models, as regards digital capability identification, based on the University course rubric dataset.

3.1 Data organisation

Since we needed some labelled samples to train and validate the machine learning models, a senior learning designer from the Learning Design and Teaching Innovation team (specialists in the application of transformative educational technologies, and are a source of knowledge for learning spaces and their capabilities from the University) manually labelled some of the collected samples. The original data was collected from course rubrics produced by course coordinators between years 2018 to 2020. In total, 1,735 samples were manually labelled for training the machine learning models – 791 samples labelled as digital, and 944 samples labelled as non-digital. Since the computer cannot directly read text (unstructured data), we used the bag-of-words technique to extract features from the course rubric dataset (i.e. assignment name, criterion title and sentence body) and applied the TF-IDF method to represent features, thus obtaining a large number of features but sparse data. The methods we used to reduce the number of features included tokenising each document, converting all characters to lowercase letters, removing punctuation and numbers, and trimming extra whitespace. However, even after these processes, the number of features exceeded 30,000, making it difficult to build a prediction model with efficiency using such a sparse, high-dimensional dataset. To reduce the number of features, we applied TF-IDF to extract key features (see Fig. 1). Instead of using the whole vocabulary (unique words) from the text, we selected only the most important features (the top *N* features, where *N* was set to 500, 800, 1,000, 1,500 and 2,000 in our experiments).

The digital capability features targeted here include ‘Information Literacy’, ‘Digital Literacy’, ‘Written Communication’, ‘Analysis’, and ‘Content’. The top features extracted from TF-IDF are considered to be closely related to these digital capabilities. As we can see from the example given in Fig. 1, features such as ‘analysis’, ‘appropriate’, ‘evidence’, ‘excellent’, ‘knowledge’, ‘referencing’, ‘research’, ‘sources’, ‘statistical’ and ‘understanding’ would be considered as indicative of digital capability within a course rubric that the machine learning models might identify and capture. In Fig. 2, a word cloud of the course rubric data is presented. We randomly shuffled the dataset and performed 5-fold cross validation. That is, we divided the dataset into five groups (each of which had the same number of samples), following which we selected one group as the testing set and used the others as the training sets. Then, we trained the machine learning models using the training sets and evaluated them using the testing set. Eventually, we calculated the average of the five results. To evaluate the performance of the machine learning models effectively, each experiment was repeated 20 times and the final results reported were based on the averages from the 20 runs.

3.2 Evaluation metrics

We used the accuracy, precision, recall and F-measure (F1-score) as performance measures and calculated these using (2)-(5), respectively. Of note, the F1-score is

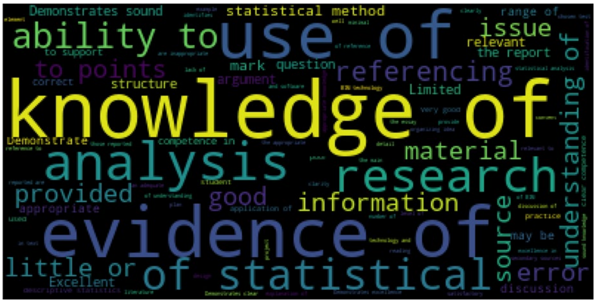


Fig. 2 A word cloud of the course rubric data

based on precision and recall and can be used to measure the overall performance of the digital capability identification models. The aim is to have higher values for the accuracy, precision, recall and F1-score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2}$$

$$Precision = \frac{TP}{TP + FP}, \tag{3}$$

$$Recall = \frac{TP}{TP + FN}, \tag{4}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{5}$$

where *TP* and *TN* are the true positives and true negatives, respectively, and *FP* and *FN* are the false positives and false negatives, respectively.

As we can see from (2)-(5), accuracy is intuitive in measuring the prediction performance with a ratio of correctly predicted samples (*TP*+*TN*) to the total number of samples (*TP*+*TN*+ *FP*+*FN*). Precision is useful for determining the performance when the cost of *FP* is high. Recall calculates how many of the actual positives (*TP*+*FP*) the models capture through labelling them as *TP*. Recall can be used to select the best model when there is a high cost associated with *FN*. The F1-score is used as a balance between precision and recall.

3.3 Parameter settings

The grid search approach is commonly used for parameter tuning, to determine the optimal parameter values for a given model by exhaustively generating a list of candidates from a grid of parameter values (Fayed & Atiya, 2019; Chiong et al., 2021). Therefore, we applied this approach with 5-fold cross validation to determine the parameter settings for all the models. Specifically, for the ANN, we optimised the number of neurons in its hidden layer (parameter values considered: [100, 200, 300 and 400]) and the maximum number of iterations (parameter values considered: [500, 600, 700, 800 and 900]). For the SVM, we optimised the regularisation parameter (parameter values considered: [100, 200, 300 and 400]) and the variance in the Gaussian kernel (Chiong et al., 2022) (parameter values considered: [0.1, 0.325, 0.55, 0.775 and 1]). For the RF and XGBoost, we optimised the number of estimators (parameter values considered: [100, 200, 300 and 400]) and the maximum depth of the trees in each estimator (parameter values considered: [4, 5, 6 and 7]).

Table 1 Experimental results for the ANN, SVM, RF, and XGBoost, based on the preprocessed digital capability dataset using 500 extracted features (best results are highlighted in bold)

Algorithm	Accuracy	Precision	Recall	F1-score
ANN	0.8254	0.8073	**0.8105**	0.8080
SVM	**0.8479**	0.8490	0.8103	**0.8284**
RF	0.7803	**0.8832**	0.5949	0.7098
XGBoost	0.8275	0.8233	0.7908	0.8060

Table 2 Experimental results for the ANN, SVM, RF, and XGBoost, based on the preprocessed digital capability dataset using 800 extracted features (best results are highlighted in bold)

Algorithm	Accuracy	Precision	Recall	F1-score
ANN	0.8127	0.7904	0.8016	0.7952
SVM	**0.8533**	0.8594	**0.8108**	**0.8338**
RF	0.7824	**0.8944**	0.5906	0.7101
XGBoost	0.8378	0.8342	0.8042	0.8179

Table 3 Experimental results for the ANN, SVM, RF, and XGBoost, based on the preprocessed digital capability dataset using 1,000 extracted features (best results are highlighted in bold)

Algorithm	Accuracy	Precision	Recall	F1-score
ANN	0.8072	0.7839	0.7973	0.7896
SVM	**0.8535**	0.8614	**0.8086**	**0.8334**
RF	0.7845	**0.9017**	0.5902	0.7119
XGBoost	0.8390	0.8398	0.7999	0.8185

Table 4 Experimental results for the ANN, SVM, RF, and XGBoost, based on the preprocessed digital capability dataset using 1,500 extracted features (best results are highlighted in bold)

Algorithm	Accuracy	Precision	Recall	F1-score
ANN	0.8134	0.7919	**0.8024**	0.7961
SVM	**0.8511**	0.8630	0.8006	**0.8298**
RF	0.7748	**0.9022**	0.5667	0.6944
XGBoost	0.8401	0.8432	0.7977	0.8190

Table 5 Experimental results for the ANN, SVM, RF, and XGBoost, based on the preprocessed digital capability dataset using 2,000 extracted features (best results are highlighted in bold)

Algorithm	Accuracy	Precision	Recall	F1-score
ANN	0.8130	0.7904	**0.8024**	0.7955
SVM	**0.8486**	0.8640	0.7924	**0.8257**
RF	0.7669	**0.8981**	0.5495	0.6801
XGBoost	0.8328	0.8338	0.7908	0.8105

3.4 Results and discussion

Tables 1, 2, 3, 4, and 5 present the experimental results for all four models using the top 500, 800, 1,000, 1,500 and 2,000 features extracted from TF-IDF, respectively.

Fig. 3 Experimental results for the ANN, SVM, RF, and XGBoost with 500 extracted features

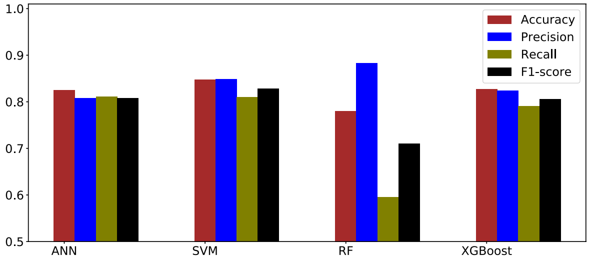
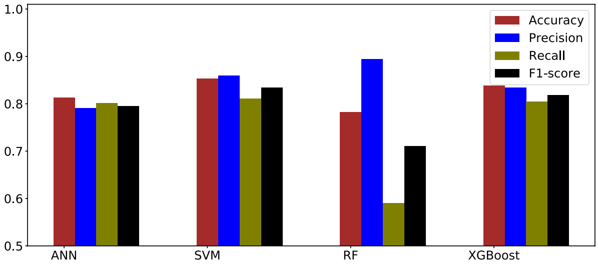


Fig. 4 Experimental results for the ANN, SVM, RF, and XGBoost with 800 extracted features



In Table 1, we see that with 500 features extracted, the SVM performs the best in terms of accuracy and F1-score. The ANN and XGBoost have similar results, while the ANN has the best recall and the RF has the best precision. Table 2 shows that the SVM, now trained with 800 extracted features, again performs the best; this time not just in terms of accuracy and F1-score but also recall. Moreover, the ANN and XGBoost outperform the RF model in terms of accuracy and F1-score. Compared with the Table 1 models (trained with 500 features), the accuracy and F1-score of all models, other than the ANN, in Table 2 improved on using more features for training. As Table 3 shows, the RF model has the best precision whereas the SVM has the best accuracy, recall and F1-score when 1,000 features are in place. Table 3 shows that on using more features to train the models, the values for accuracy and the F1-score of SVM, RF and XGBoost are higher compared with the corresponding Table 2 values, indicating improved performance. Table 4 shows that the ANN has the best recall; SVM has the best accuracy and F1-score; RF has the best precision with 1,500 features used for training. Moreover, the values for accuracy and the F1-score of ANN and XGBoost are higher compared with the Table 3 values, which indicate further improvement in performance. Table 5 again shows that the ANN has the best recall; SVM has the best accuracy and F1-score; RF has the best precision. However, a comparison with the corresponding values in Table 4 shows that the performance of all the models in Table 5 has deteriorated in terms of accuracy and F1-score. Thus, Tables 1, 2, 3, 4, and 5 show that on including more features for training, the performance of machine learning models improves at first but then deteriorates on including too many features. Among all the models in all the tables, the SVM performs the best in terms of accuracy and F1-score.

In addition, to better visualise the results in Tables 1, 2, 3, 4, and 5, we present Figs. 3, 4, 5, 6 and 7, which graphically illustrate the accuracy, precision, recall and

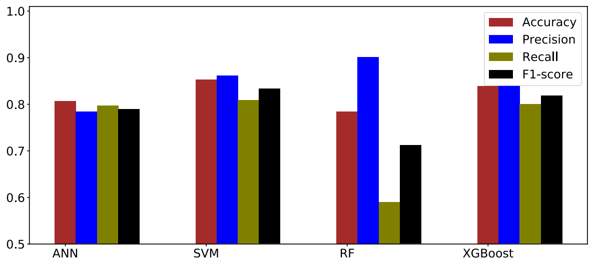


Fig. 5 Experimental results for the ANN, SVM, RF, and XGBoost with 1,000 extracted features

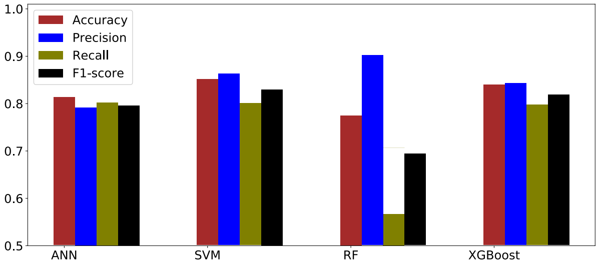


Fig. 6 Experimental results for the ANN, SVM, RF, and XGBoost with 1,500 extracted features

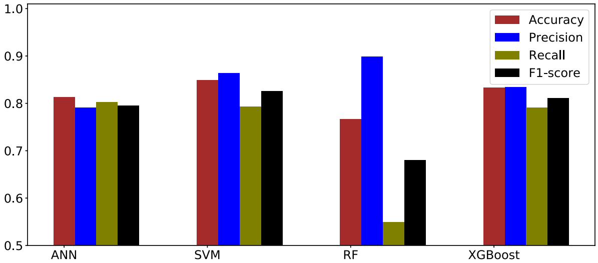


Fig. 7 Experimental results for the ANN, SVM, RF, and XGBoost with 2,000 extracted features

F1-score of the models. As the figures show, the SVM outperforms others in terms of accuracy and F1-score for all the cases with different features tested.

In summary, on using more features for training, the model performance increases at first and then decreases, which means including an appropriate number of features in the prediction models is important. The SVM performs better than the ANN, RF and XGBoost in terms of accuracy and F1-score on including varying numbers of extracted features. The ANN ranks first in terms of performance when 500 extracted features are used, whereas the SVM, RF and XGBoost have the best F1-score when 800, 1,000 and 1,500 extracted features are used, respectively. Our experimental results confirmed that the prediction models can effectively identify digital capabilities in courses across the University.

3.5 Statistical analysis

Although the results presented thus far indicated that the SVM performs the best for digital capability identification, it is unclear whether the differences between

Table 6 Statistical test results for the ANN, SVM, RF, and XGBoost, based on the pre-processed digital capability dataset using 500 extracted features (*p*-values less than 0.05 are highlighted in bold) in terms of F1-score

	SVM	RF	XGBoost
ANN	**6.9977**$\times 10^{-5}$	**6.3018**$\times 10^{-8}$	0.7251
SVM		**6.3018**$\times 10^{-8}$	**4.4160**$\times 10^{-5}$
RF			**6.3018**$\times 10^{-8}$

Table 7 Statistical test results for the ANN, SVM, RF, and XGBoost, based on the pre-processed digital capability dataset using 800 extracted features (*p*-values less than 0.05 are highlighted in bold) in terms of F1-score

	SVM	RF	XGBoost
ANN	**1.5340**$\times 10^{-7}$	**6.3018**$\times 10^{-8}$	**1.7006**$\times 10^{-5}$
SVM		**6.3018**$\times 10^{-8}$	**1.0639**$\times 10^{-3}$
RF			**6.3018**$\times 10^{-8}$

Table 8 Statistical test results for the ANN, SVM, RF, and XGBoost, based on the pre-processed digital capability dataset using 1,000 extracted features (*p*-values less than 0.05 are highlighted in bold) in terms of F1-score

	SVM	RF	XGBoost
ANN	**2.7545**$\times 10^{-7}$	**6.3018**$\times 10^{-8}$	**2.4450**$\times 10^{-5}$
SVM		**6.3018**$\times 10^{-8}$	**2.0444**$\times 10^{-5}$
RF			**6.3018**$\times 10^{-8}$

Table 9 Statistical test results for the ANN, SVM, RF, and XGBoost, based on the pre-processed digital capability dataset using 1,500 extracted features (*p*-values less than 0.05 are highlighted in bold) in terms of F1-score

	SVM	RF	XGBoost
ANN	**1.2856**$\times 10^{-6}$	**6.3018**$\times 10^{-8}$	**2.1678**$\times 10^{-5}$
SVM		**6.3018**$\times 10^{-8}$	**0.0305**
RF			**6.3018**$\times 10^{-8}$

the results obtained are statistically significant. To address this issue, we report the results of the statistical tests we conducted based on the Wilcoxon rank-sum test (Murakami, 2015) to verify the significance. We ran the tests 20 times.

Tables 6, 7, 8, 9 and 10 show the statistical test results based on the experimental results from the 20 runs in terms of F1-score. In these tables, *p*-values greater than the significance level of 0.05 are highlighted in bold. From the tables, we can conclude that all results between two different models are

Table 10 Statistical test results for the ANN, SVM, RF, and XGBoost, based on the pre-processed digital capability dataset using 2,000 extracted features (p-values less than 0.05 are highlighted in bold) in terms of F1-score

	SVM	RF	XGBoost
ANN	$\mathbf{3.7358}\times 10^{-6}$	$\mathbf{6.3018}\times 10^{-8}$	$\mathbf{2.4450}\times 10^{-5}$
SVM		$\mathbf{6.3018}\times 10^{-8}$	$\mathbf{1.2866}\times 10^{-3}$
RF			$\mathbf{6.3018}\times 10^{-8}$

significantly different, except the pair of ANN and XGBoost with 500 features extracted for training.

In addition, we would like to ascertain whether the differences between the results obtained for the same model on using different numbers of features are statistically significant or not. Table 11 presents the results of the statistical tests on the experimental results for the same model in terms of F1-score, based on the Wilcoxon rank-sum test. From the table, several observations can be made:

(1) For the ANN, its model performance on including 500 extracted features exceeded the performance on including more features, in terms of F1-score.
(2) For the SVM, although the results showed that it has the best performance on including 800 extracted features for training, the differences between the results obtained for the models are not significant.
(3) For the RF, although the model that includes 1,000 extracted features has the best F1-score, the differences between the results obtained for this model and for the models using 500 and 800 extracted features are not significant.
(4) For the XGBoost, although its model based on 1,500 extracted features has the best F1-score, only the differences between the results for this model and the model using 500 extracted features are significant.

In summary, although each model of ANN, SVM, RF and XGBoost has the best F1-score using 500, 800, 1,000 and 1,500 extracted features, respectively, in future studies, it is sufficient to include 500 extracted features for the ANN, SVM and RF and 800 extracted features for XGBoost, considering the performance and efficiency (with less features extracted, the training process can be faster). The results

Table 11 Statistical test results for the ANN, SVM, RF, and XGBoost based on different extracted features in terms of F1-score (p-values less than 0.05 are highlighted in bold)

#Features	ANN	SVM	RF	XGBoost
500	*	0.1762	0.7455	$\mathbf{6.8303}\times 10^{-3}$
800	**0.0173**	*	0.7868	0.9138
1000	$\mathbf{1.8661}\times 10^{-3}$	0.8924	*	0.9784
1500	$\mathbf{5.7959}\times 10^{-3}$	0.4328	$\mathbf{5.7959}\times 10^{-3}$	*
2000	$\mathbf{8.6940}\times 10^{-3}$	0.1231	$\mathbf{8.7721}\times 10^{-5}$	0.0659

#Features means the number of features and ‘*’ stands for the best result for the model in a given column confirmed that the SVM can significantly outperform the other models in terms of F1-score.

4 Conclusions

In this digital age, it is vital for university students to master certain digital knowledge. Identifying digital capabilities in university courses is thus an important endeavour. However, since about 4,000 courses were offered across the University of Newcastle, Australia, a manual analysis of assessment rubrics was deemed infeasible, and machine learning was used instead. In this study, we analysed the performance of machine learning models for digital capability classification based on real-world university course rubric data using various numbers of features extracted from the course rubrics. The experimental results showed that, when using more features for training, the model performance increases initially and then decreases, which means the inclusion of an appropriate number of features in the prediction models is essential.

With the high prediction accuracy, the machine learning-based prediction models can be used to automatically identify courses with digital capabilities. This, in turn, would allow the University to gain a better understanding about where and what digital capabilities are formally assessed in the courses, and design and map curriculums to develop the digital capabilities of both students and staff. Based on the classification results, educators can develop proper strategies – through curriculum design and mapping – to help equip students with the necessary skills for their future career development. Course coordinators could also identify what digital capabilities are missing in certain courses. Professional staff who need specific digital capacity training can be recommended to undertake certain courses; students could also select the courses with the digital capabilities of their interest. In addition, based on the prediction results, some related courses could be automatically recommended to the students based on their current digital capabilities and related majors.

In future work, we plan to continue our collaboration with the Learning Design and Teaching Innovation team at the University of Newcastle, Australia, to further develop this automated machine learning approach by tailoring it to their specific needs and requirements. Technically speaking, despite the SVM having the highest accuracy and F1-score, there is room for further improvement. One option is to explore the use of fuzzy weights (Fan et al., 2020; 2022) to alleviate the influence of noise data by evaluating the importance of different samples, considering the fact that most of the off-the-shelf machine learning-based models studied here do not have the capability to do so. In addition, although there were about 4,000 courses, only 1,735 samples were labelled in this study. We will attempt to label more data to further improve the performance of the prediction models, or explore clustering-based methods so that we do not need to rely on labelled data.

Acknowledgements Zongwen Fan would like to acknowledge financial support from the University of Newcastle, Australia.

Author contributions Zongwen Fan: Conceptualization, Methodology, Investigation, Formal analysis, Software, Validation, Writing - original draft. Raymond Chiong: Conceptualization, Methodology, Investigation, Supervision, Resources, Writing - review & editing.

This was a university project; Raymond Chiong was contacted by the university and initiated the idea of this work. Zongwen Fan was employed as a research assistant, carried out the experiments, and drafted the manuscript. Raymond Chiong provided guidance and edited the manuscript. Both authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions The authors would like to acknowledge support from Learning Design and Teaching Innovation team at the University of Newcastle, Australia. The first author's research is also supported by the Scientific Research Funds of Huaqiao University (No. 21BS122).

Data availability All data included in this study is available from the first author and can also be found in the manuscript.

Code availability All code included in this study is available from the first author upon reasonable request.

Declarations

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate Not applicable.

Consent for Publication Not applicable.

Competing interests The authors declare that they have no competing interests.

Conflict of interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Balyan, R., McCarthy, K.S., & McNamara, D.S. (2020). Applying natural language processing and hierarchical machine learning approaches to text difficulty classification. *International Journal of Artificial Intelligence in Education*, *30*(3), 337–370.

Bishop, C.M. (2006). *Pattern recognition and machine learning*. New York: springer.

Borges, A.F., Laurindo, F.J., Spínola, M. M., Gonçalves, R. F., & Mattos, C.A. (2020). The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions. *International Journal of Information Management*, p. 102225.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Brownlee, J. (2017). Deep learning for natural language processing: develop deep learning models for your natural language problems. *Machine Learning Mastery*.

Burges, C.J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Carmona, P., Climent, F., & Momparler, A. (2019). Predicting failure in the us banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, *61*, 304–323.

Chen, T., He, T., Benesty, M., Khotilovich, V., & Tang, Y. (2015). Xgboost: extreme gradient boosting. R package version 0.4-2, pp. 1–4.

Chiong, R., Fan, Z., Hu, Z., & Chiong, F. (2021). Using an improved relative error support vector machine for body fat prediction. *Computer Methods and Programs in Biomedicine*, *198*, 105,749.

Chiong, R., Wang, Z., Fan, Z., & Dhakal, S. (2022). A fuzzy-based ensemble model for improving malicious web domain identification. *Expert Systems with Applications*, p. 117243. https://doi.org/10.1016/j.eswa.2022.117243.

Crosby, A., Pham, K., Peterson, J.F., & Lee, T. (2020). Digital work practices: Affordances in design education. *International Journal of Art & Design Education*, *39*(1), 22–37.

Edwards, R., & Fenwick, T. (2016). Digital analytics in professional work and learning. *Studies in Continuing Education*, *38*(2), 213–227.

Elphick, M. (2018). The impact of embedded ipad use on student perceptions of their digital capabilities. *Education Sciences*, *8*(3), 102.

Fan, Z., Chiong, R., & Chiong, F. (2022). A fuzzy-weighted gaussian kernel-based machine learning approach for body fat prediction. *Applied Intelligence*, *52*, 2359–2368.

Fan, Z., Chiong, R., Hu, Z., & Lin, Y. (2020). A fuzzy weighted relative error support vector machine for reverse prediction of concrete components. *Computers & Structures*, *230*, 106,171.

Fayed, H.A., & Atiya, A.F. (2019). Speed up grid-search for parameter selection of support vector machines. *Applied Soft Computing*, *80*, 202–210.

Friedman, J.H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Gislason, P.O., Benediktsson, J.A., & Sveinsson, J.R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, *27*(4), 294–300.

González-Rojas, O., Correal, D., & Camargo, M. (2016). Ict capabilities for supporting collaborative work on business processes within the digital content industry. *Computers in Industry*, *80*, 16–29.

Krasuska, M., Williams, R., Sheikh, A., Franklin, B.D., Heeney, C., Lane, W., Mozaffar, H., Mason, K., Eason, S., Hinder, S., & et al. (2020). Technological capabilities to assess digital excellence in hospitals in high performing health care systems: International edelphi exercise. *Journal of Medical Internet Research*, *22*(8), e17,022.

Land, W.H., & Schaffer, J.D. (2020). The support vector machine. In *The art and science of machine intelligence* (pp. 45–76). Springer.

Murakami, H. (2015). The power of the modified wilcoxon rank-sum test for the one-sided alternative. *Statistics*, *49*(4), 781–794.

Pagani, L., Argentin, G., Gui, M., & Stanca, L. (2016). The impact of digital skills on educational outcomes: Evidence from performance tests. *Educational Studies*, *42*(2), 137–162.

Qader, W.A., Ameen, M.M., & Ahmed, B.I. (2019). An overview of bag of words; importance, implementation, applications, and challenges. In *2019 International Engineering Conference (IEC)* (pp. 200–204). IEEE.

Qaiser, S., & Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, *181*(1), 25–29.

Tang, M., & Liao, H. (2021). Multi-attribute large-scale group decision making with data mining and subgroup leaders: An application to the development of the circular economy. *Technological Forecasting and Social Change*, *167*, 120,719.

Whetstone, D., & Moulaison-Sandy, H. (2020). Quantifying authorship: A comparison of authorship rubrics from five disciplines. *Proceedings of the Association for Information Science and Technology*, *57*(1), e277.

Wilson, C.B., & Slade, C. (2020). Developing digital capabilities of future students through consensus curriculum development. *ETH Learning and Teaching Journal*, *2*(2), 292–295.

Zhong, Z.J. (2011). From access to usage: The divide of self-reported digital skills among adolescents. *Computers & Education*, *56*(3), 736–746.

Zhu, E., Chen, Y., Ye, C., Li, X., & Liu, F. (2019). Ofs-nn: An effective phishing websites detection model based on optimal feature selection and neural network. *IEEE Access*, *7*, 73,271–73,284.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.