 DOI: 10.31276/VJSTE.64(4).10-15

Applying Bayesian neural network to evaluate the influence of specialized mini projects on final performance of engineering students: A case study

Minh Truong Nguyen[1], Viet-Hung Dang[2], Truong-Thang Nguyen[2*]

[1]*University of Sciences, Vietnam National University, Hanoi*

[2]*Faculty of Building and Industrial Construction, Hanoi University of Civil Engineering*

Received 10 June 2022; accepted 8 September 2022

Abstract:

In this article, deep learning probabilistic models are applied to a case study on evaluating the influence of specialized mini projects (SMPs) on the performance of engineering students on their final year project (FYP) and cumulative grade point average (CGPA). This approach also creates a basis to predict the final performance of undergraduate students based on their SMP scores, which is a vital characteristic of engineering training. The study is conducted in two steps: (i) establishing a database by collecting 2890 SMP and FYP scores and the associated CGPA of a group of engineering students that graduated in 2022 in Hanoi; and (ii) engineering two deep learning probabilistic models based on Bayesian neural networks (BNNs) with the corresponding architectures of 8/16/16/1 and 9/16/16/1 for FYP and CGPA, respectively. The significance of this study is that the proposed probabilistic models are capable of (i) providing reasonable analysis results such as the feature importance score of individual SMPs as well as an estimated FYP and CGPA; and (ii) predicting relatively close estimations with mean relative errors from 6.8% to 12.1%. Based on the obtained results, academic activities to support student progress can be proposed for engineering universities.

Keywords: **data, engineering, machine learning, neuron network, project.**

Classification numbers: **1.3, 2.3**

Introduction

Nowadays, universities are capable of collecting data with reference to their students in electronic format. As a result, there is an urgent need to effectively transform large volumes of data into knowledge to improve the quality of managerial decisions and to predict academic performance of students at an early stage. As a part of artificial intelligence (AI) techniques recently adopted in a wide variety of human life applications [1, 2], various machine learning (ML) approaches have been increasingly applied to analyse educational data, such as student scores, to concentrate academic assistance on students as well as to improve the university training programs. ML is an especially appealing alternative in the field of engineer training and education as it is difficult or unfeasible to develop conventional algorithms to perform required tasks [3, 4].

S.S. Abu-Naser, et al. (2015) [4] developed an artifical neural network (ANN) model for predicting student performance at the Faculty of Engineering and Information Technology, Al-Azhar University, based on the registration records of 1407 students using a feed forward back propagation algorithm for training. The model was tested with an overall result of 84.6%. E.Y. Obsie, et al. (2018) [5] developed a neural network model for predicting student cumulative grade point averages for the 8th semester (CGPA8) and designed an application based on the predictive models. The real dataset employed in the study was gathered from 134 students at the Hawassa University School of Computer Science that graduated in 2015, 2016, and 2017. It is shown that the student progress performance, which is measured by CGPA8, can be predicted using scores from their first-, second-, and third-year courses. Z. Iqbal, et al. (2017) [6] utilized collaborative filtering (CF), matrix factorization (MF), and restricted Boltzmann machine (RBM) techniques to systematically analyse real-world data collected from 225 undergraduate students enrolled in the Electrical Engineering program at the Information Technology University (ITU) from which the academic performance of the ITU students was evaluated. It was shown that the RBM technique was

*Corresponding author: Email: thangnt2@huce.edu.vn

better than the other techniques in predicting student performance in the particular course. S.D.A. Bujang, et al. (2021) [7] introduced a comprehensive analysis of machine learning techniques to predict final student grades in first semester courses by improving the performance of predictive accuracy. The performance accuracy of six well-known machine learning techniques, namely, decision tree (J48), support vector machine (SVM), naïve bayes (NB), K-nearest neighbour (K-NN), logistic regression (LR), and random forest (RF) using 1282 student course grades were presented and followed by a multiclass prediction model to reduce the over-fitting and misclassification results caused by imbalanced multi-classification using the Synthetic Minority Oversampling Technique (SMOTE) with a two feature selection method. It was shown that the proposed model integrated with RF had significant improvement with the highest f-measure of 99.5% [7].

It is worth mentioning that most of the aforementioned ML approaches were conducted in a deterministic manner. Hence, there is a need to develop a probabilistic model that is capable of providing well-predicted results as well as estimating the confidence of the results through associated intervals. Such a result is more relevant for experimental data on student exam scores rather than a single point estimation because, even with the same student, scattered results can be obtained from different series of experiments.

In training programs at engineering universities, specialized mini projects (SMPs) play an important role as they progressively provide knowledge as well as accumulate conceiving, designing, implementing, and operating skills necessary for their FYP, which is an integrated topic to solve a practical problem of the particular field of engineer training. This article applies a machine learning approach to predict FYP and final CGPA results from those SMPs based on which the influence of SMPs on the FYP and CGPA can be evaluated in a data-driven manner. A case study is conducted by collecting 2890 datapoints in the form of score results from eight SMPs, one FYP, and the CGPA of a group of 289 engineering students that graduated in 2022 in Hanoi. Then, two deep learning probabilistic models based on BNNs are established for FYP and CGPA predictions. It is shown from the obtained results that the proposed approach is a practical tool providing quick and reasonable analysis results such as the feature importance score of an individual SMP and the estimated FYP and CGPA results. Furthermore, a relatively close estimation can be captured from the BNN model for CGPA, providing useful information for academic management.

Stochastic model using BNNs

As pointed out by various authors, a major obstacle to the data-driven method is the scarcity of relevant data, and this problem becomes accentuated when studying the obtained scores of various individual students [8-12]. Even with data in hand, there exist unavoidable deviations between them [13-14]. Thus, this study proposes to engineer a probabilistic machine learning model on the basis of BNNs rather than deterministic ones as done in the reviewed works. The advantage of such a probabilistic model is that it is capable of predicting quantities of interest such as the FYP score or CGPA, as well as estimating the amount of uncertainty that is associated with the prediction values. It is evident that the more data available, the more accurate the model, and vice versa. In summary, the key contribution of this article is to propose a probabilistic ML model to predict the FYP score and CGPA results from the given scores of SMPs so that the effect of an individual SMP as well as FYP on the final performance of students can be evaluated.

We begin by briefly reviewing the ANN to set up mathematical symbols and terminology. Given a dataset $D=[X,Y]$, the i[th] data sample is denoted by $X_i=[x_{i,1},\dots,x_{i,n}]$ with n being the number of features. Herein, each feature is an input related to SMP and FYP results. It is desirable to develop a non-linear mapping from X to Y, i.e., $Y_i=f(X_i)$. A standard architecture of ANN consists of an input layer, an output layer, and one or many hidden layers with the total number of layers of the ANN being L. Each layer consists of various neurons that are fully connected with all neurons in neighboring layers. Mathematically, a neuron j at layer l could be described by a linear transformation plus a non-linear activation function, as follows:

$$x_j^l = h\left(\sum_{k=1}^{N_{l-1}} w_{j,k}^l\, x_k^{l-1} + b_j^{l-1}\right) \quad (1)$$

where x_j^l is output of neuron j at layer l; x_k^{l-1} is output of neuron k at the previous layer l-1; $w_{j,k}^l$ is a weight assigned to the connection between the former and the latter; N_{l-1} is the total number of neurons of layer l-1; b_j^{l-1} is a real value to be determined, also known as bias; and h is an activation function. Here, the sigmoid activation function is used to squeeze values into the range (0,1). Moreover, the function is continuous and differentiable everywhere, thus rendering the training process of the neural network via a gradient descent-based algorithm faster and more effective.

By setting W^l as the matrix of weights corresponding to layer l, with l=1,…,L, the ANN could be described by the following equation derived from the description of neural networks in [15]:

$$\hat{Y}_i = F(X_i|W) = f_L(\dots(f_2(f_1(X_i|\mathrm{W}^1)|\mathrm{W}^2)\dots|\mathrm{W}^L) \quad (2)$$

where $\hat{Y}_i$ is a prediction of Y_i, and f_l with l=1,…,L denotes transformation operations at layer l in the ANN. The network will be iteratively trained to determine the optimal values of W^l that minimize the discrepancy between $\hat{Y}_i$ and Y_i.

BNN is a probabilistic deep learning model that combines the high prediction performance of ANN with the ability to estimate uncertainty of the Bayes theory [16]. In the authors' opinion, the model is especially suitable for working with not-so-abundant collected data owing to two reasons: (i) in practice, similar series of experiments with identical input parameters still provide different results due to unavoidable uncertainty; and (ii) fitting an ANN with many parameters to a limited database may cause the over-fitting problem, i.e., ANN is likely to yield low-accuracy results on new data despite being well trained. In other words, it is necessary to not only perform prediction of FYP and CGPA results but also to estimate how much confidence we have about the prediction results.

For this purpose, rather than assigning the deterministic values for weight W of the neural network, BNN uses a Gaussian probability distribution for W as below:

$$W = \mu + \sigma \times \epsilon \ \ with\ \epsilon \sim \mathcal{N}(0, I) \tag{3}$$

where μ and σ denote the mean and standard deviation matrices of W and ϵ is the noise drawn from a zero-mean unity-variance normal distribution. Then, μ, σ are parameters to determine through the learning process.

Note that the output of BNN is a probability distribution, thus a specialized loss function L is required to measure the model's performance. The adopted metric is the Kullback-Leibler divergence (KL) whose formula is:

$$L = KL(q(W|\mu,\sigma)||p(W|D)) = E_{q(W|\mu,\sigma)}\, log\, \frac{q(W|\mu,\sigma)}{p(W|D)} \tag{4}$$

Next, the optimal values of μ^* and σ^* are the solutions of the following minimization problem:

$$\mu^*, \sigma^* = \underset{\mu,\sigma}{\operatorname{argmin}}\, KL(q(W|\mu,\sigma)||p(W|D)) \tag{5}$$

Via the Bayes' rule, $p(W|D)$ can be calculated as below:

$$p(W|D) = \frac{p(D|W)p(W)}{p(D)} \tag{6}$$

Substituting Eq. (6) into Eq. (4), the loss function L is rewritten as follows:

$$L = E_{q(W|\mu,\sigma)} \log p(D) - E_{q(W|\mu,\sigma)} \log p(W) - E_{q(W|\mu,\sigma)} \log p(D|W) \tag{7}$$

This loss function can be approximated from observed discrete data as follows:

$$L = \frac{1}{N_s}\sum_{i=1}^{N_s} [\log q(W_i|\mu,\sigma) - \log p(W_i) - \log p(D|W_i)] \tag{8}$$

where N_s is the total number of samples.

Next, the gradients of the loss function with respect to µ and σ are derived by:

$$\Delta\mu = \frac{\partial L(W|\mu,\sigma)}{\partial W} + \frac{\partial L(W|\mu,\sigma)}{\partial \mu}$$
$$\Delta\sigma = \frac{\partial L(W|\mu,\sigma)}{\partial W} \times \epsilon + \frac{\partial L(W|\mu,\sigma)}{\partial \sigma} \tag{9}$$

Finally, μ and σ are updated using a small learning rate α as follows:

$$\sigma \leftarrow \sigma - \alpha \times \Delta\sigma; \ \ \mu \leftarrow \mu - \alpha \times \Delta\mu. \tag{10}$$

Case study - Database on score results of SMPs, FYP and CGPA

In the postgraduate training program of civil engineers, a student is required to pass all the subjects including eight specialized mini projects before being qualified to conduct his/her final year project. The SMPs consist of: Design of Architecture (DoA), Design of Foundation (DoF), Mechanism of Reinforced Concrete Structures (MoRCS); Design of Reinforced Concrete Buildings (DoRCB); Design of Structural Steel Buildings (DoSSB); Construction Technology 1 (CT1); Construction Technology 2 (CT2); and Construction Management (CM) that will be numbered from SMP.1 to SMP.8, respectively. Each individual SMP provides students with the corresponding knowledge and professional skill that will be integrated in his/her FYP to design and build a civil/industrial building in real situation. It is noteworthy that the SMPs' and FYP's exams are all conducted in the form of oral defence. As a result, all the aforementioned SMPs significantly influence the FYP, which together with all theoretical subjects and SMPs contributes to the CGPA (Fig. 1).

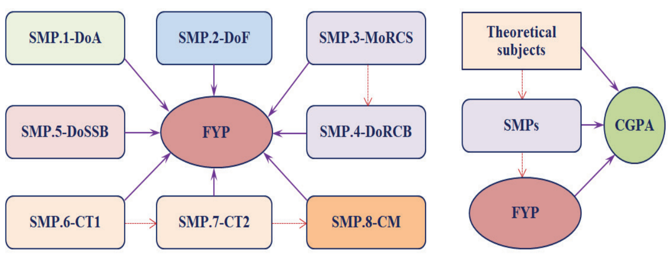


Fig. 1. Projects in training program of civil engineers in HUCE.

In this research, a dataset of 2890 scores of eight SMPs, one FYP, and the CGPA is collected from 289 civil engineers who graduated from Hanoi University of Civil Engineering (HUCE) in 2022. Figs. 2-5 display the histograms of all the score results of their SMPs, FYP, and CGPA on the 4-point scale, showing clear visualization of the range of values as well as their distributions.

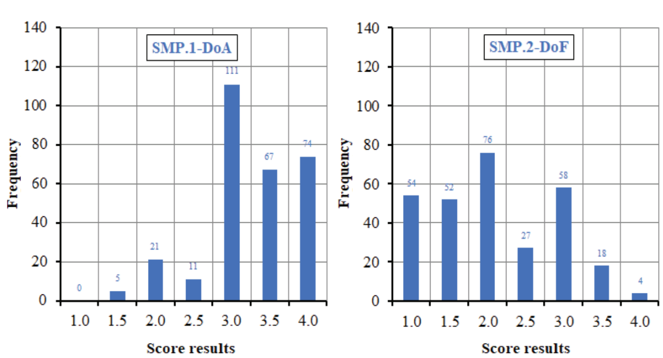


Fig. 2. Score histograms of SMP.1-DoA and SMP.2-DoF.

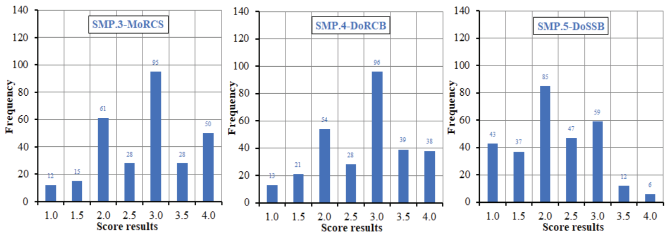


Fig. 3. Score histograms of SMP.3-MoRCS, SMP.4-DoRCB and SMP.5-DoSSB.

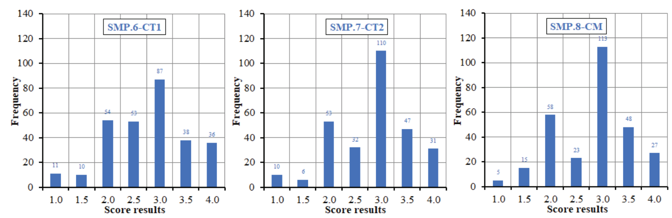


Fig. 4. Score histograms of SMP.6-CT1, SMP.7-CT2 and SMP.8-CM.

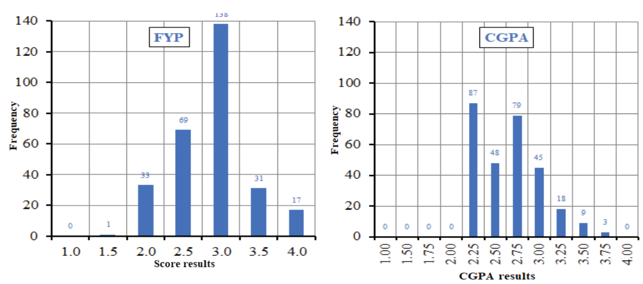


Fig. 5. Histograms of FYP and CGPA results.

It can be seen in Fig. 2 that the score results of SMP.1-DoA of the investigated 289 students were quite high, among which 74 students obtained 4.0, whereas the biggest group was 3.0 with 111 students. There were 67 students that earned a 3.5 and the number of students receiving scores of 2.5, 2.0, and 1.5 were 11, 21, and 5, respectively. It is noted that although this is the first SMP of engineering students in the university program, there is not much calculation in this task of architectural design. The remaining seven SMPs are all critical to students as they are curtailed to train them on the design as well as construction of buildings. It can be observed in the distributions of these SMP scores that the group of 3.0 is almost dominant, except the cases of SMP.2-DoF and SMP.5-DoSSB, of which the dominant group was 2.0 (Figs. 2 and 3). It is shown in Figs. 3 and 4 that the distribution of the main SMPs in the program were similar to each other. Meanwhile, it can be observed in Fig. 5 that the distribution of FYP results was quite standard, whereas there is a regression trend of the number of students with the higher scores in the CGPA results.

Analysis results on FYP and CGPA using BNN model

In this study, the adopted architecture of the BNN for FYP scores is 8/16/16/1. The model consists of an input layer with 8 neurons, 2 hidden layers with 16 neurons, and an output layer with one neuron corresponding to the FYP result, as graphically illustrated in Fig. 6A. The neurons in the input layer correspond to the score results of SMP.1 to SMP.8. Since the data size is moderate, it is reasonable to avoid using too deep architectures of many hidden layers as well as wide layers with a significant number of neurons, which may lead to a pronounced increment of parameters to determine. Besides, the number of neurons for the hidden layer is set to 16 since it should be a power of 2 to be convenient for the memory of the computer. For the proposed BNN model, each neuron has two parameters to be determined, characterizing the probability distribution of its weight. At the beginning of learning, they are initialized as a normal distribution with zero mean and unity standard deviation. For prediction of CGPA, the corresponding BNN architecture is 9/16/16/1 in which the FYP score in the dataset is combined with the SMPs as the 9th element of the input layer (Fig. 6B).

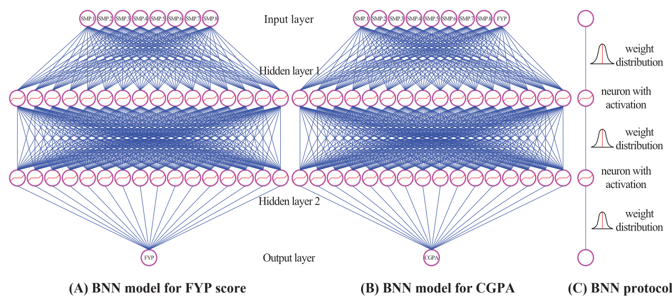
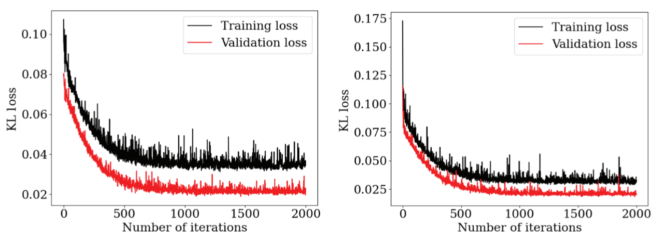


Fig. 6. Graphical representation of BNN whose weights are characterized by probability distributions.

Updating the model's weights as described in Eq.(10) is based on the Adam Optimization Algorithm belonging to the first-order gradient descent optimization family, which gradually adapts the model's weights by a small amount after each iteration to reduce the loss function. The number of updates is controlled through a hyper-parameter of 0.001. This value can also be referred to as the learning rate that was determined via a preliminary test to ensure the learning process is convergent within a reasonable learning time. It is noted that a small learning rate will unnecessarily increase learning time, whereas a large value could lead to premature results. On the other hand, pre-processing standardization is adopted to obviate the scale difference issue between different features with different physical meanings. By utilizing the aid of the deep learning library Pytorch for building the overall framework, the deep probabilistic library Pyro [17] for establishing the BNN-based data-driven model, Pandas for data management, scikit-learn library [18] for data standardization, and Matplotlib for data visualization, the implementation of the proposed data-driven framework can be realized in this study.

In this study, the investigated database is split into three non-overlapping datasets, namely, the training, validation, and testing sub-sets with a ratio of 60:20:20, corresponding to data from 173, 58, and 58 students.

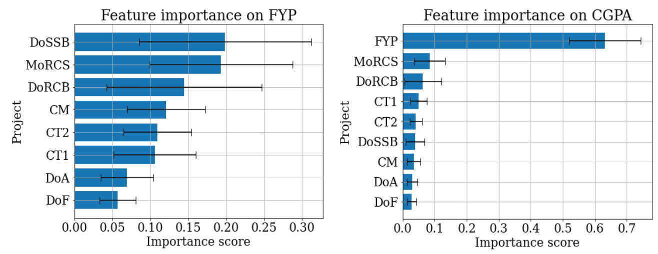
After being built, the probabilistic models were trained with the training database mentioned in the previous section. Figs. 7A and 7B depict the learning curves of the BNN models of FYP and CGPA results, respectively, showing how loss functions evolve versus the number of training iterations (epochs) on both training and validation datasets. In the following paragraphs, the corresponding results of the BNN model for CGPA will be given in parentheses. The KL loss function of FYP (CGPA) quickly dropped for epochs from 0 to 500, before gradually decreasing to values around 0.035 and 0.02 (0.03 and 0.025) on the training and validation datasets, respectively. After that, a steady trend is observed, i.e., no clear improvement is obtained, until the number of epochs reached 2000. For epochs 1300 to 1400, there is less fluctuation in loss function than the other intervals. Hence, an iteration value of 1300 is selected as the final configuration of the BNN model.



(A) BNN model for FYP (B) BNN model for CGPA

Fig. 7. Evolution of the KL loss function against number of iterations on training and validation datasets for (A) FYP and (B) CGPA.

The effect of each SMP on FYP and CGPA results can be investigated via a feature importance study. Importance scores will be assigned for all SMPs, a high score means that the corresponding SMP has a significant impact on the FYP/CGPA and a low score means there is less impact. The SMPs are ranked based on their importance score. Such results provide understanding about the correlation between the projects and helps students optimize their learning strategy to achieve desired final scores. The permutation feature importance method is used along with the proposed data-driven model. Herein, a feature refers to an SMP score. The simple, yet effective, core idea of this method is to permute the values of features and evaluate the change in prediction errors. A permuted feature means that original values of this feature are shuffled among data samples while other features remain unchanged. If a permuted feature incurs large errors, this feature is important and contributes significantly to the prediction results. Hence, all features will be permuted one by one, and the respective error will be calculated for each case. Next, these errors are sorted in descending order, and importance scores are derived. Since the BNN is a probabilistic model, the feature importance results obtained with BNN are random variables. Thus, one repeats the feature importance calculations 100 times and then derives their statistical characteristics such as mean, min, and max values. The influence of SMPs on a student's final performance on their FYP and CGPA in terms of importance score are shown in Fig. 8A and Fig. 8B, respectively.

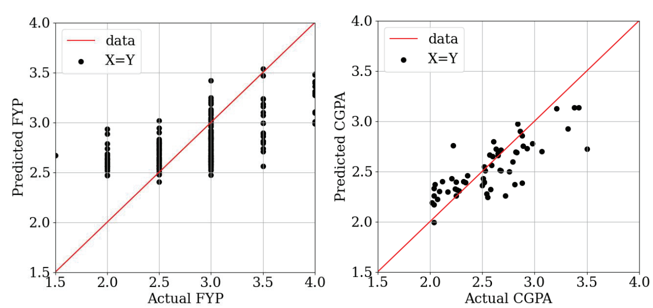


(A) BNN model for FYP (B) BNN model for CGPA

Fig. 8. Feature importance graph of SMPs' influence on (A) FYP and (B) CGPA.

It can be observed from Fig. 8A that, compared to the remaining SMPs, the mini projects of DoSSB and MoRCS have more pronounced influence on FYP results. It is noted that in the training program the SMP and FYP have 1 and 10 credits, respectively (each credit equals to 15 hours of lecturing). It can be seen in Fig. 8B that the influence of the FYP on CGPA, which is counted from a total of 168 credits, is most significant while MoRCS holds the second position. The large gap between the feature importance scores of FYP and MoRCS is reasonable due to their credit relative ratio of 10:1. Fig. 8 also proves that reinforced concrete and steel structures are the leading applications with significant distributions currently in the field of construction.

Next, the final performance of the trained model is evaluated on the test dataset and the prediction results of FYP and CGPA are demonstrated in Fig. 9.



(A) BNN model for FYP (B) BNN model for CGPA

Fig. 9. Prediction results of (A) FYP scores and (B) CGPA.

It is shown in Fig. 9 that for each data point (shown in the dot symbol), its X-coordinate denotes true FYP scores from the database while the Y-coordinate is a value predicted by the model. Ideally, a perfect model will provide the same results as those from the database, as highlighted by the red 45-degree line. While the BNN model for FYP is not as close as expected in Fig. 9A, the predicted points of CGPA lie relatively close to the ideal line in Fig. 9B. Specifically, the mean relative errors of predicted FYP and CGPA results are 12.1% and 6.8%, respectively. These results qualitatively confirm the viability of the proposed BNN model for CGPA.

Conclusions

This study applied a data-driven method for assessing the FYP score and CGPA based on the results of eight SMPs, which are considered as a characteristic of the training program for engineering students. Probabilistic machine learning models based on the BNN were introduced and the theoretical foundation of the model and key parameters of the proposed approach were described, followed by a case study using a database of 2890 score results of SMPs, FYP, and CGPA from a group of civil engineers that graduated in 2022 in Hanoi. One of the main results of the proposed model is that it can be utilized to evaluate the influence of an individual SMP on a student's final performance in terms of FYP and CGPA. It was shown from the results of the case study that the subjects commonly applied in practice also contribute a more significant influence. In addition, the BNN model for CGPA is capable of providing relatively close predictions with a mean relative error of 6.8%. Furthermore, the application of the data-driven model is straightforward as it is built based on open source libraries and the user-friendly programming language, Python, without requiring any specialized software.

For the next step of the study, other models such as straight artificial neural network and drop-out neural network can also be incorporated for comparison purposes. Furthermore, one can complement the database with the score results of all students that graduated before, during, or after the COVID-19 pandemic to gain an overall picture to propose appropriate solutions to improve academic activities of engineering universities, where the projects play very important role for undergraduate students.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] T. Mitchell (1997), *Machine Learning*, McGraw-Hill Education, 432pp.

[2] J. Hu, H. Niu, J. Carrasco, B. Lennox, F. Arvin (2020), "Voronoi-based Multi-robot autonomous exploration in unknown environments via deep reinforcement learning", *IEEE Transactions on Vehicular Technology*, **69(12)**, DOI: 10.1109/TVT.2020.3034800.

[3] Y. Bengio, Y. LeCun, G. Hinton (2015), "Deep learning", *Nature*, **521(7553)**, pp.436-444.

[4] S.S. Abu-Naser, I.S. Zaqout, M. Abu-Ghosh, R.R. Atallah, E. Alajrami (2015), "Predicting student performance using artificial neural network: In the Faculty of Engineering and Information Technology", *International Journal of Hybrid Information Technology*, **8(2)**, pp.221-228.

[5] E.Y. Obsie, S.A. Adem (2018), "Prediction of student academic performance using neural network, linear regression and support vector regression: A case study", *International Journal of Computer Applications*, **180(40)**, pp.39-47.

[6] Z. Iqbal, J. Qadir, A.N. Mian, F. Kamiran (2017), "Machine learning based student grade prediction: A case study", https://arxiv.org/abs/1708.08744.

[7] S.D.A Bujang, A. Selamat, R. Ibrahim, O. Krejcar, E. Herrera-Viedma, H. Fujita, N.A.M. Ghani (2021), "Multiclass prediction model for student grade prediction using machine learning", *IEEE Access*, **9**, pp.95608-95621.

[8] S.T. Jishan, R.I. Rashu, N. Haque, R.M. Rahman (2015), "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique", *Decis. Analyst*, **2(1)**, pp.1-25.

[9] A. Polyzou, G. Karypis (2016), "Grade prediction with models specific to students and courses", *Int. J. Data Sci. Analyst*, **2(3-4)**, pp.159-171.

[10] I. Khan, A. Al Sadiri, A.R. Ahmad, N. Jabeur (2019), "Tracking student performance in introductory programming by means of machine learning", *Proc. 4th MEC Int. Conf. Big Data Smart City (ICBDSC)*, pp.1-6.

[11] M.A. Al-Barrak, M. Al-Razgan (2016), "Predicting students final GPA using decision trees: A case study", *Int. J. Inf. Educ. Technol.*, **6(7)**, pp.528-533.

[12] E.C. Abana (2019), "A decision tree approach for predicting student grades in research project using WEKA", *Int. J. Adv. Comput. Sci. Appl.*, **10(7)**, pp.285-289.

[13] F. Ahmad, N.H. Ismail, A.A. Aziz (2015), "The prediction of students' academic performance using classification data mining techniques", *Appl. Math. Sci.*, **9**, pp.6415-6426.

[14] T. Anderson, R. Anderson (2017), "Applications of machine learning to student grade prediction in quantitative business courses", *Glob. J. Bus. Pedagog.*, **1(3)**, pp.13-22.

[15] Ng. Andrew (2019), *Stanford University CS229: Lecture Notes*, 216pp.

[16] C.M. Bishop (1997), "Bayesian neural networks", *Journal of the Brazilian Computer Society*, **4**, pp.61-68.

[17] E. Bingham, et al. (2019), "Pyro: Deep universal probabilistic programming", *The Journal of Machine Learning Research*, **20(1)**, pp.973-978.

[18] F. Pedregosa, et al. (2011), "Scikit-learn: Machine learning in Python", *The Journal of Machine Learning Research*, **12**, pp.2825-2830.