Journal of Information Systems Engineering and Management
2025, 10(24s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Educational Data Analysis and Classification using Deep Learning Techniques

Mr. SuginLal. G[1], Dr. R. Porkodi[2]
[1]*Research Scholar, Centre for Research and Evaluation, Bharathiar University, Coimbatore - 46, Tamil Nadu, India*
[2]*Professor, Department of Computer Science, Bharathiar University, Coimbatore - 46, Tamil Nadu, India*

ARTICLE INFO

Received: 26 Dec 2024
Revised: 15 Feb 2025
Accepted: 25 Feb 2025

ABSTRACT

In today's cutthroat society, it is essential for an institution to predict student performance, classify people according to their talents, and work to improve their performance in upcoming exams. To increase academic achievement, students should be instructed to focus their efforts in a particular area well in advance.This research propose novel method in educational data analysis based on academic student data performance using deep learning techniques by feature extraction as well as classification. Input has been collected as academic student performance data and processed for noise removal, normalization and dimensionality reduction. The processed data features has been extracted utilizing kernel quantum based reward Q-neural network and classified using ensemble VGG-19 with encoder_ convolutional architecture. Various student academic performance data are subjected to experimental analysis in terms of accuracy, precision, recall, F-1 score, RMSE, and MAP. Proposed method achieved 99% accuracy, 98% precision, 99% recall, a 98% F-1 score, 0.002% RMSE, and 99% MAP.

Keywords: educational data analysis, academic student data, deep learning, Q- neural network, encoder_ convolutional architecture.

1. Introduction:

The conversion of inputs, such as human as well as financial resources, into outputs occurs through a complex process in the educational activity. By way of comparison, labour as well as capital inputs employed by a school are likely to have an impact on its output, similar to type of production function that is often utilized to analyse method of a factory.A basic framework, however, falls short of accurately capturing some of the most important conspicuous aspects of the process because students themselves are both an input as well as an output, and they themselves are transformed by experience of education. This problem is well-known in the literature that is currently available on the Educational Production Function (EPF) [1]. Students' learning processes are, in fact, influenced by their personal traits as well as those of their family, peers, the community in which they reside, and the features of the school they attend. Additionally, the way that different inputs translate into output is probably going to differ significantly depending on educational methods that are in place in different nations. Establishing the hierarchy's structure is a difficult task, not least since the structure may vary between nations [2]. A logical way to comprehend how, all other things being equal, the differences between educational systems can affect students' results is through exploring international datasets that contain data about students' performance in more countries. Based on these variables, ML classification methods are employed to forecast student performance [3]. The research on machine learning for education uses diverse datasets and evaluation criteria to assess the effectiveness of categorization models. Most popular classification methods in literature—DT, NB, ANN, SVM and RF were evaluated and their performances were compared [4]. Additionally, different students perform differently and have variable levels of comprehension. Additionally, it might be challenging to collect and analyse data on kids' various educational needs [5]. Recently, ML techniques have been used to forecast students' academic achievement, especially during their first year in college. Main goal of ML is to find hidden and important correlations in data with a variety of properties. Based on information contained in university's database, several machine learning algorithms, such as decision trees, are proven to be efficient for predicting performance of students. With aim of automating process of predicting student results, a machine learning-based forecast of academic student performance was created [23]. Artificial Intelligent such as SARIMA, multi-variable regression, ridge regression, and KNN regression for prediction water level [6].

Contribution of this research is as follows:

1. Topropose novel technique in educational data analysis based on academic student data performance utilizing DL methods by feature extraction and classification.
2. Input has been collected as academic student performance data and processed for noise removal, normalization and dimensionality reduction.
3. Processed data features has been extracted utilizing kernel quantum based reward Q- neural network and classified using ensemble VGG-19 with encoder_ convolutional architecture.

2. Related works:

A variety of areas, including forecasting student performance, extracting data for decision-making, and exploring relationships, are included in the research and application of ML in educational methods[7]. Machine learning techniques [8] were used to anticipate how well the students will perform in helping the learners. The study involved 117 master's students, and NB and ANN produced the greatest results when compared to well-known classification algorithms. An ICRM2 algorithm was applied to 419 high school students and employed a variety of learning methods to estimate the student dropouts [9]. For a data set of 772 students, author [10] employed early projections of institute's students, relying on students' academic data, efficacy, and video interaction. The study of intelligent computation techniques and their applications [11] is extensive. The examination of data mining methods and strategies that could be used with an intelligent computer system was also considered [12]. A descriptive and preventive statistical research employing data mining methods was provided [13] to forecast student performance in the Bronze Capital. In order to identify students who are "high risk" of failing course, a study was conducted to compare ML methods utilized to predict student performance on tests [14]. A methodology that tests eight ML algorithms to forecast students' academic achievement in a course was proposed [15]. In a fresh data set from a private university in United Arab Emirates (UAE), an RF methodwas employed to predict students' academic achievement [16]. A system for forecasting and categorising student performance using five different machine learning algorithms was proposed in [17]. Early intervention (with varied meanings of "early") is one of the most effective techniques for keeping students in school [18]. When an intervention is linked to an academic skill or attendance, retention may increase by as much as 13% [19]. Although some of the methodologies 50 that can be employed with learning analytics tools are utilized to support early interventions, this is not their intended application. [20] proposes a model utilizing ensemble learning and multiparametric analysis to enhance student classification accuracy. Effectiveness of Student Evaluation of Teaching Effectiveness (SETE) [21] test was examined using statistical methods, NN, and Bayesian data reduction techniques. The findings do not support the use of SETE as a broad barometer of online learning or teaching efficacy. A decision support method[22] was proposed for a tutor to forecast students' performance.

3. System model:

This section discusses a cutting-edge method for classifying and extracting features using deep learning approaches to analyse educational data based on academic student achievement. Here, the input has undergone noise removal, normalisation, and dimensionality reduction processing after being acquired as academic student performance data.Kernel quantum based reward Q- NN was utilized to extract features from processed data, and ensemble VGG-19 with encoder_ convolutional architecture was used to classify it. In figure 1, the suggested architecture is illustrated.

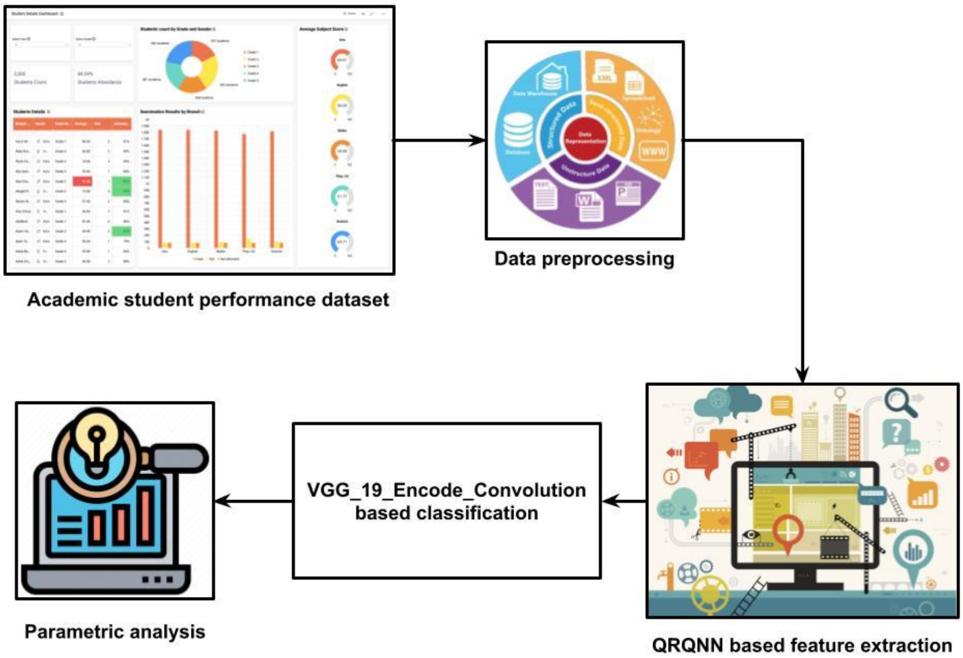

Figure1: Proposed System

(i) Pre-processing:

Data preparation is used as the first stage of data processing. It changes material into a format that the ML algorithms can use. Data gathered from UCI with multivariate attributes is pre-processed. The pre-processing procedures are followed below.

a. Importing Libraries: Pandas, label encoder, train test split, RF classifier, and accuracy score were imported. Each class's label is stored in the label encoder. It uses an ordinal or one-hot encoding approach to encode categorical attributes. Data are divided into a training set and a test set using train test split function. A test is provided with a set of 30% of the data to be utilized. In order to obtain repeated results, the random seed was supplied via the random state argument. The DT classifiers that were used on various data subsets to create RF classifier. The results of various classifiers are averaged, which increases predicted accuracy and reduces over-fitting. The amount of matches between actual as well as anticipated values is represented by the accuracy score.

(b) Handling of missing values:A dataset may occasionally have "holes" in it known as missing values. Regression analysis is one statistical technique that cannot be used on a dataset with missing values.To create results that are useful, these observations must be eliminated or replaced using the proper statistical techniques. The missing values can be found using pandas in Python; pandas will identify both empty cells and "NA" kinds as missing values. To find the missing values, we utilised the dataset isnull().any().sum() method [13].

(c) Splitting data: Data is divided into test and training sets for machine learning. Records with well-known labels are included in training set. Training dataset is utilized to train ML method, which helps it develop learning capabilities. The model's propensity for prediction is next examined using test data with unidentified labels. The partitioning was done using the train test split () method.In this study, a 70–30 split was used, where 70% of data made up training dataset and remaining 30% test dataset.

(ii) Feature Extraction using Kernel quantum based Q-neural network:

When a quantum system has a finite number of qubits and is utilised to solve a classical problem, it is typically necessary to treat the classical data in a way known as dimensionality reduction first. After performing these procedures, we are left with a 1 × m2 vector, which we can then translate into angle information α by using the equation 1,

$$\alpha = \pi \mathrm{x} \qquad (1)$$

where α = [α1, α2 . . . αm2]. The QNN is a quantum circuit within a sequence of parameter-dependent quantum gates (unitary operators) which act on quantum input data. Generally, a QNN can be shown as equation 2,

$$\mathrm{U}(\theta) = \prod_{\mathrm{l}=1}^{\mathrm{N}} \mathrm{V_l} \mathrm{U_l}(\theta_\mathrm{l}) \qquad (2)$$

which is a product of N quantum layers. The l-th quantum layer consists of product of non-parametric quantum gates Vl and parametric quantum gates Ul(θl) where θl are variational parameters. We can further represent the parametric quantum gates Ul(θl) in l-th layer as the production of S parametric quantum gates by equation (3),

$$U_l(\theta_l) \equiv \otimes_{j=1}^{S} U_{l,j}\left(\theta_{l,j}\right) \qquad (3)$$

In which each parametric quantum gate Ul,j (θl,j) can be transformed with Euler's formula as equation (4),

$$\exp\left(-i\theta_{l,j}P\right) = I\cos\left(\theta_{l,j}\right) - i\sin\left(\theta_{l,j}\right)P \qquad (4)$$

where i is the imaginary number, I is a 2 × 2 identity matrix, and P is a Pauli operator from the set {X, Y , Z} that acts on qubits. The output of the QNN is the measurement result on a computational basis of the readout qubits. Since the measurement result of a qubit is probabilistic, the expectation value E of the measurement results is used as the QNN output by equation (5),

$$E = \left\langle \Psi_x \middle| U^{\dagger}(\theta) M U(\theta) \middle| \Psi_x \right\rangle \qquad (5)$$

where |Ψxi is the input quantum state of the QNN and M is a linear combination of Pauli operators that serve as observables for readout qubits. The loss L of a training sample in hybrid quantum-classical model is calculated in conventional manner on a classical device. For the given training sample, the loss L is calculated with an objective function `(.) of the task on the expected output y and the actual output Eusing equation (6),

$$\overline{\mathrm{L} = \ell(\mathrm{E}, \mathrm{y})} \qquad (6)$$

During the model optimization phase, like in the classical NN, back-propagation and gradient descent will be performed to update variational parameters in the QNN. The gradient ofa variational parameter θ_k in k^{th} quantum layer with respect to with respect to loss L is calculated by equation (7)

$$\frac{\partial L}{\partial \theta_k} = \frac{\partial L}{\partial E}\frac{\partial E}{\partial \theta_k} \qquad (7)$$

It is easy to obtain ∂L/∂E according to the objective function $\ell(), .\partial\dot{E}/\,\partial\theta_k$could be calculated by equation (8) and (9),

$$\frac{\partial E}{\partial \theta_k} = i\left\langle \Psi_x \middle| U_-^{\dagger}\left[P_k, U_+^{\dagger} H U_+\right] U_- \middle| \Psi_x \right\rangle \qquad (8)$$

$$U_+ = \prod_{l=k+1}^{N} V_l U(\theta_l) \text{ and } U_- = \prod_{l=1}^{l=k} V_l U(\theta_l) \qquad (9)$$

With the gradients of parameters, the classical device sets updated parameters for QNN using an optimization algorithm, such as SGD. An alternative gradient calculation for a quantum model is parameter-shift, which obtains the gradients by running the same VQCs with shifted parameters and calculating the difference in their outputs. The architecture overview of the proposed design is shown in figurer 2.

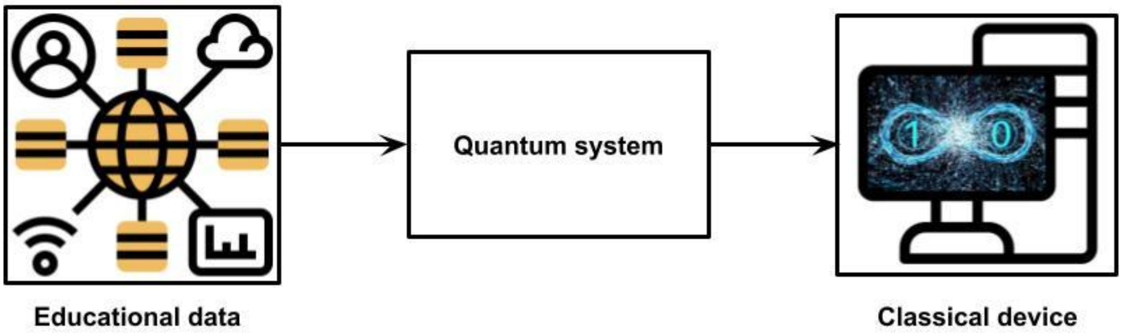

Figure 2: QNN Architecture

F is a potential function if there exists a real-valued function $\phi: S\ddot{\mathbb{R}}$ such that for all $s \in S - \{s_0\}, a \in \hat{\imath}, s' \in S$using equations (10-15),

$$F(s, s') = \gamma\phi(s') - \phi(s) \quad (10)$$

$$F(s, a, s', a') = \gamma\phi(s', a') - \phi(s, a) \quad (11)$$
$$F(s, t, s', t') = \gamma\phi(s', t') - \phi(s, t) \quad (12)$$
$$F(s, a, t, s', a', t') = \gamma\phi(s', a', t') - \phi(s, a, t) \quad (13)$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + F + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (14)$$

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}[\textstyle\sum_{t=0}^{\infty} \gamma^t r(s_t)], \quad (15)$$

where$s_0 \sim \rho_0(s_0), a_t \sim \pi(a_t \mid s_t), s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$

Value function Vπ, advantage function Aπ, and state action value function Qπ will all be defined using the following conventions by equations (16–19):

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}} \ldots [\textstyle\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})], \quad (16)$$
$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots}[\textstyle\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})], \quad (17)$$
$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s), \text{ where} \quad (18)$$
$$a_t \sim \pi(a_t \mid s_t), s_{t+1} \sim P(s_{t+1} \mid s_t, a_t) \text{ for } t \geq 0 \quad (19)$$

The following important identity describes how the advantage over another strategy π, expressed in terms of expected return, is built over timesteps by equation (20):

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}}[\textstyle\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)]\Big]_1 \quad (20)$$

where $\mathbb{E}_{s_0, a_0, \cdots \sim \pi}[\ldots]$denotes that actions are sampled $a_t \sim \tilde{\pi}(\cdot \mid s_t)$. Let ρ_π be discounted visitation frequencies calculated using equation (21)

$$\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \cdots \quad (21)$$

where s0 ~ ρ0 and actions are selected using π. eqn (21) can be revised by replacing the timesteps with a sum over states in equations (22-24),

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_{t=0}^{\infty}\sum_{s} P(s_t = s \mid \tilde{\pi}) \sum_{a} \tilde{\pi}(a \mid s)\gamma^t A_\pi(s,a) \quad (22)$$
$$= \eta(\pi) + \sum_{s}\sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid \tilde{\pi}) \sum_{a} \tilde{\pi}(a \mid s) A_\pi(s,a) \quad (23)$$
$$= \eta(\pi) + \sum_{s} \rho_\pi(s) \sum_{a} \tilde{\pi}(a \mid s) A_\pi(s,a) \quad (24)$$

However, in the approximate situation, it will often be inevitable that some states s for which expected advantage is negative, that is$\sum_{a} \tilde{\pi}(a \mid s) A_\pi(s,a) < 0$exist. This is because estimate and approximation mistake. It is challenging to directly optimise Equation (25), given the complicated dependence of $\rho_\pi(s)$ on $\tilde{\pi}$. Instead, we present local approximation η to as follows:

$$L(\tilde{\pi}) = \eta(\pi) + \sum_{t=0}^{\infty}\sum_{s} P(s_t = s \mid \tilde{\pi}) \sum_{a} \tilde{\pi}(a \mid s) \quad (25)$$

$L\pi$ ignores variations in state visiting density brought on by changes in policy and instead uses visitation frequency instead of visitation density. But if we have a parameterized policy πθ, $L\pi$ matches to η first order if πθ(a|s) is a differentiable function of parameter vector θ.

Algorithm of reward Q-NN:

$\forall s, a Q(s,a) = 0$
s ← start state
maxepisode reward ← $-\infty$
min episodereward← ∞
replay buffer ← null
while True do

Episode True do

Episode reward=0

While s is not terminal do

a ← Choose an action based on a policy and given state r, s' ← do action, get reward and observe next state episode reward ← episode reward $+r$
if $r = 0$ then
| $\phi_s \leftarrow 0$ else $\phi_s \leftarrow 1 + \frac{\text{episode reward-max episode reward}}{\text{max episode reward-min episode reward}}$
append replay buffer (s, a, r, ϕ_s, s')
s ← s'
if update period or s is terminal then

$i \leftarrow$ length of replay buffer -1 to 0 by -1 do
$s, a, r, \phi_s, s' \leftarrow$ replay buffer [i]
$F_{s'}, - \leftarrow \leftarrow$ replay buffer [i + 1]
$Q \leftarrow \gamma\phi_{s'} - \phi_s$
$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + F + \gamma \max_a Q(s',a) - Q(s,a) \right]$

End

Replay buffer is null

End

End

If episode reward> max episode reward then

Max episode reward← episode reward

Else if

Episode reward < min episode reward
Then
Min episode reward ← episode reward
End
End

(iii) Classification using VGG-19_Encode_convolutional network:

The convolutional layer's filters are typically 3X3 trainable feature extractors. Following the stack of convolutional layers, the max-pooling operation and ReLU activation function are used. ReLU is currently most well-liked nonlinear activation function, defined as part of its argument that is positive when x is input to a neuron by equation (26):

$$f(x) = \max(0, x) \qquad (26)$$

ReLU function is computationally more effective than sigmoid function, has higher convergence performance, and resolves the vanishing gradient issue. Each neuron in a dense layer receives information from all neurons present in layer below it, making it a completely linked layer. It is necessary to specify activation function for layer with dense connections.

By dropping out a random set of activations in that layer and setting them to zero, the Dropout layer compels network to be redundant. The overfitting issue is mitigated by randomly removing neurons during training phase. This layer is never utilised during testing; only during training.

An appropriate model is selected to understand the fundamental organisation of CNN feature vectors of images. In order to achieve this, a sparse AE is employed, a symmetrical NN mostly utilised for unsupervised feature learning from data sets. Sparse AE typically consists of encoding and decoding components, as shown in equation (27),

$$\hat{\boldsymbol{h}}_i = g\left(\mathbf{W}^{(e)}\boldsymbol{x}_i + \boldsymbol{b}^{(e)}\right) \qquad (27)$$

The encoder weight matrix is $\boldsymbol{W}^{(e)} \in \mathfrak{H}^{S\times D}$ and the encoding bias vector is $\boldsymbol{b}^{(e)} \in \mathfrak{H}^{s}$. The sigmoid function, or g vð Þ¼ 1=ð Þ 1 þ expð Þ v, is a common option for the activation function. In order to recreate the input vector xi as shown in equation (28) during decoding phase, hidden representation is also mapped through a non-linear activation function.

$$\hat{\boldsymbol{x}}_i = g\left(\mathbf{W}^{(d)}\boldsymbol{h}_i + \boldsymbol{b}^{(d)}\right) \qquad (28)$$

where$\mathbf{W}^{(d)} \in^{D\times s}$ is decoding weight matrix, $\boldsymbol{b}^{(d)} \in^{D}$ is decoding bias, and x^i is reconstructed CNN feature vector. Input vector must be mapped by encoder to hidden layer before receiving a new feature expression. Equation (29) represents function.

$$y = f(x) = s\left(W^{(1)}x + b^{(1)}\right) \qquad (29)$$

Decoder's job is to trace hidden layer's expression y back to original input. Equation (30) represents function.

$$x = g(y) = s\left(W^{(2)}y + b^{(2)}\right) \qquad (30)$$

Thus, reconstruction error for every data is given by equation (31)

$$L = \| x - g(f(x)) \|^2 \qquad (31)$$

Cost function is described using equation (32),

$$J(W,b) = \left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}\left\|x^{(i)} - g\left(f\left(x^{(i)}\right)\right)\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}}\left(W_{ji}^{(i)}\right)^2 \quad (32)$$

Here x stands for original input data, x1 for corrupted input data, y for newly created feature that was gained by encoding x1, and z for result of decoding y. Equation (33) provides reconstruction error,

$$L_D = \left\|x - g\left(f(x_1)\right)\right\|^2 \quad (33)$$

Cost function is given by equation (34)

$$J_D(W,b) = \left[\frac{1}{m}\sum_{i=1}^{N}\left(\frac{1}{2}\left\|x^{(i)} - g\left(f\left(x_1{}^{(i)}\right)\right)\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}}\left(W_{ji}^{(l)}\right)^2 \quad (34)$$

Typically, all that is required to produce x1 is to randomly set units in x to zero in accordance with noise figure ($k \in [0, 1]$). The parameters are solved using the same process as the autoencoder. Spectrum, texture, shape, and other types of information are all included in the input data block. Without the need for manual feature extraction, SADE can implicitly learn these features as well as apply them for categorization.

The image block has more information with a larger S, which makes classification easier. The results of the categorization could be impacted by a variety of items in an image block, though, if S is too large. Every image block's label is a vector with a dimension equal to the total number of categories (n). Only possible values for each node in the vector are 0 and 1. The vector's m^{th} number is set to 1 and the others are set to 0 if the picture block falls under the m^{th} category.

4. Performance analysis:

This section compares the different methodologies and summarises the experimental findings. Python and PyCharm were used to carry out the implementation on a Windows 10 computer.

Dataset description: Two educational datasets are utilized to assess performance of classification methods: one from Kaggle via xAPI [3, 4] and one from the UCI repository, both of which contain data on Portuguese students. Details of experimental datasets are displayed in Table 1.

Table 1. Specifications of datasets utilized in experiments.

Dataset	Features	Observations	Class labels
Student academic performance dataset-portuguese	32	1044	Excellent Good Satisfactory Sufficient Fail
Student academic performance dataset -xAPI	16	480	Excellent Good Satisfactory Sufficient Fail

Table 2: Comparative analysis of proposed and existing techniquesoverEducational data

Dataset	Techniques	Accuracy	Precision	Recall	F1_Score	RMSE	MAPE
UCI repository	ANN	94	90	90	90	02	99
	SETE	87	80	78	79	04	97
	Quantum-VGG	99	97	98	97	003	99
Kaggle by xAPI	ANN	84	88	73	78	05	96
	SETE	76	84	59	63	07	95
	Quantum-VGG	99	99	99	99	002	99

Based on student data analysis in educational analysis, Table 2 compares and contrasts suggested and current techniques. Here, the characteristics that were examined included recall, accuracy, precision, F-1 score, RMSE, and MAP. UCI repository and Kaggle by xAPI datasets are being compared.

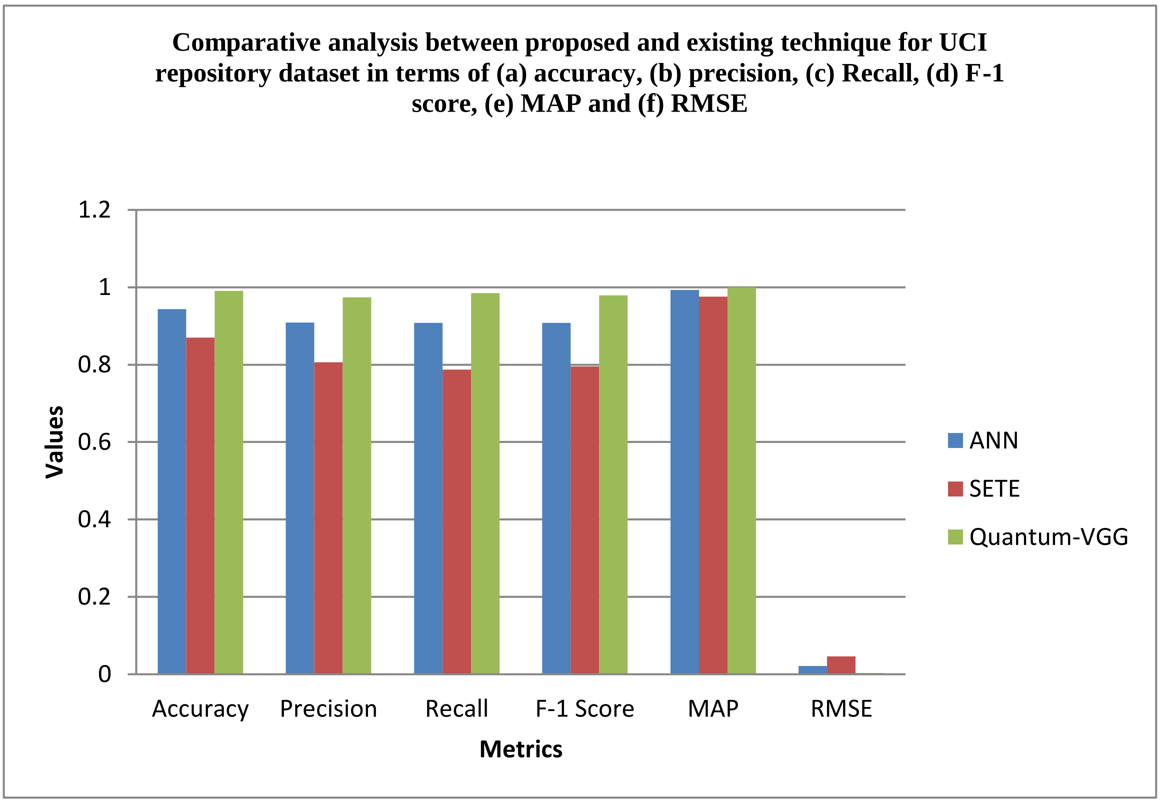

Figure 3: Comparative analysis between proposed and existing technique for UCI repository dataset in terms of accuracy, precision, Recall, F-1score,MAP and RMSE

For the dataset from the UCI repository, Figure 3 compares the proposed and existing techniques. The proposed method achieved 99% accuracy, 97% precision, 98% recall, 97% F-1 score, 003% RMSE, and 99% MAP.

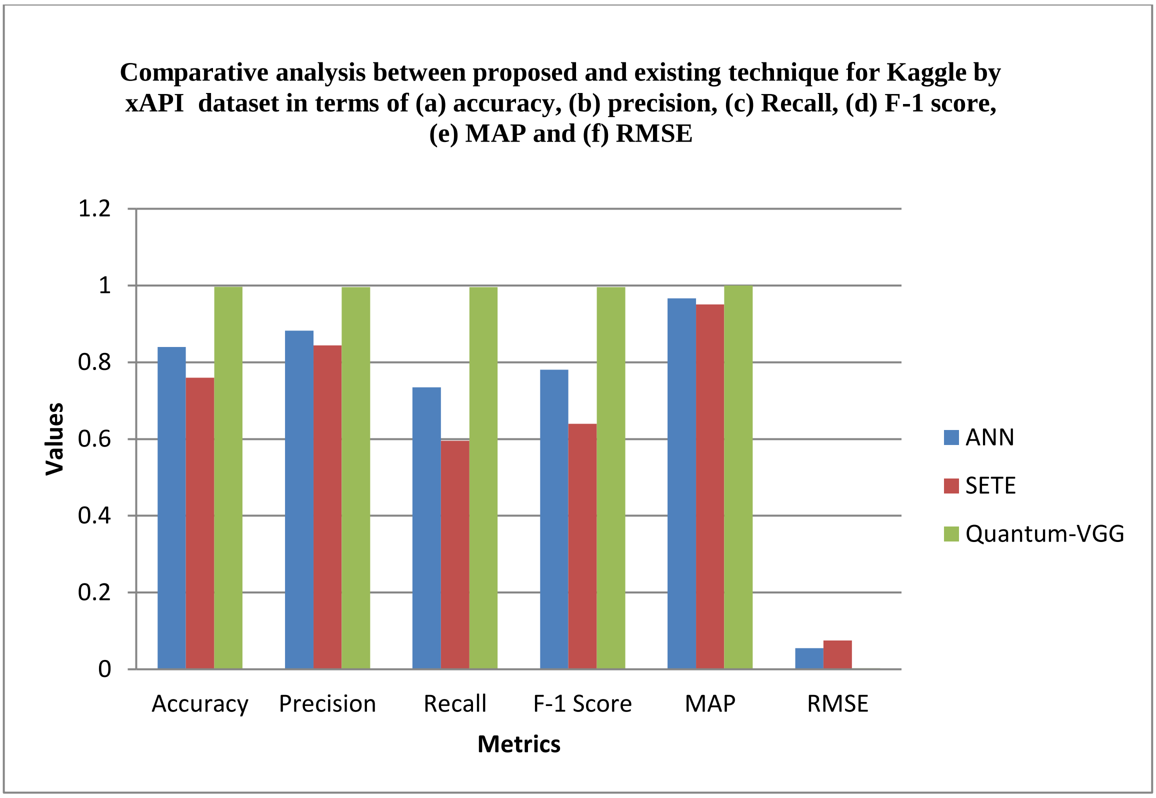

Figure 4: Comparative analysis between proposed and existing technique for Kaggle by xAPI dataset in terms of accuracy, precision, Recall, F-1 score, MAPand RMSE

Based on a comparison between the proposed and existing techniques, as shown in figure 4, the suggested technique achieved accuracy of 99%, precision of 99%, recall of 99%, F1 score of 99%, RMSE of 0.002%, and MAP of 99% for the Kaggle by xAPI dataset.

Advances in information processing technologies have been made possible through deep learning and knowledge discovery. The exceptional outcomes of the evaluation processes performed a critical and irreplaceable role throughout education. AI systems can extract, transform, and extract data, find the hidden conceptual norms and relationships that are useful to our article on the overall group situation, and facilitate final decision-making based on the classification technique. They can also summarise and analyse data from a wide range of complex information. The college English online teaching model can benefit from features like automatic homework correction, online question and answer functionality, speech recognition evaluation, and data collection and analysis in the learning process. Deep learning is closely related to the realisation and optimization of these functions. These functions, which are crucial to college English online teaching method, can be continuously optimised using the deep learning technique that this paper proposes combines clustering algorithm and neural network. Four advanced graduate students with a background in educational methods were enlisted for experiment. The participants were split evenly between two men and two women. In a survey made up of 10 post-reply pairs, participants were asked to respond to three questions: (1) "Do you think the reply was generated by a human or machine?" (2) "Did the reply provide informational, emotional, and community support?" and (3) "How good was the quality of the reply in terms of grammars, readability, and coherence to the discussion context?". Participants might choose more than one sort of support for a response if appropriate, or they could choose none if no support could be clearly shown. Participants received a definition and examples of three categories of assistance prior to experiment.

5. Conclusion:

This method presents an innovative approach to analysing educational data by leveraging deep learning techniques for feature extraction and classification of academic student performance. Utilizing a kernel quantum-based reward

Q-neural network for feature retrieval and an ensemble VGG-19 with encoder_convolutional architecture for data classification, the model is designed to predict the student outcomes using parameters such as midterm exam grades, departmental data, and faculty data. The results of this data-driven research are instrumental in shaping a framework for learning analysis in higher education and aiding decision-making processes. Significantly, the study identifies the most efficient machine learning methods and contributes to the early prediction of students at a high risk of failing. The proposed method demonstrates impressive performance metrics, achieving 99% accuracy, 98% precision, 99% recall, 98% F-1 score, 0.002% RMSE, and 99% MAP.

Reference:

[1] Wu, J., Tao, Z., & Li, Q. (2022). Scalable Quantum Neural Networks for Classification. *arXiv preprint arXiv:2208.07719*.

[2] Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(1), 1-19.

[3] Shafiq, D. A., Marjani, M., Habeeb, R. A. A., &Asirvatham, D. (2022). Student Retention using Educational Data Mining and Predictive Analytics: A Systematic Literature Review. *IEEE Access*.

[4] Dabhade, P., Agarwal, R., Alameen, K. P., Fathima, A. T., Sridharan, R., &Gopakumar, G. (2021). Educational data mining for predicting students' academic performance using machine learning algorithms. *Materials Today: Proceedings*, *47*, 5260-5267.

[5] Guleria, P., &Sood, M. (2022). Explainable AI and machine learning: performance evaluation and explainability of classifiers on educational data mining inspired career counseling. *Education and Information Technologies*, 1-36.

[6] Maltare, N. N., Sharma, D. & Patel, S. (2023). An Exploration and Prediction of Rainfall and Groundwater Level for the District of Banaskantha, Gujrat, India. International Journal of Environmental Sciences, 9(1), 1-17. https://www.theaspd.com/resources/v9-1-1-Nilesh%20N.%20Maltare.pdf

[7] Veluri, R. K., Patra, I., Naved, M., Prasad, V. V., Arcinas, M. M., Beram, S. M., & Raghuvanshi, A. (2022). Learning analytics using deep learning techniques for efficiently managing educational institutes. *Materials Today: Proceedings*, *51*, 2317-2320.

[8] Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., &Phasinam, K. (2021). Classification and prediction of student performance data using various machine learning algorithms. *Materials today: proceedings*.

[9] Yousafzai, B. K., Khan, S. A., Rahman, T., Khan, I., Ullah, I., Ur Rehman, A., ... &Cheikhrouhou, O. (2021). Student-performulator: student academic performance using hybrid deep neural network. *Sustainability*, *13*(17), 9775.

[10] Rupasi, R., Jwalitha, T., Sudarshan, R., Manasa, T., &Gundabatini, G. (2021). Analysis of Educational Data Mining Using Machine Learning Algorithms. *Journal of Xi'an University of Architecture & Technology*.

[11] Hussain, S., & Khan, M. Q. (2021). Student-performulator: predicting students' academic performance at secondary and intermediate level using machine learning. *Annals of data science*, 1-19.

[12] Albreiki, B., Zaki, N., &Alashwal, H. (2021). A systematic literature review of student'performance prediction using machine learning techniques. *Education Sciences*, *11*(9), 552.

[13] Okewu, E., Adewole, P., Misra, S., Maskeliunas, R., &Damasevicius, R. (2021). Artificial neural networks for educational data mining in higher education: A systematic literature review. *Applied Artificial Intelligence*, *35*(13), 983-1021.

[14] Patil, J. M., & Gupta, S. R. (2021). Extracting Knowledge in Large Synthetic Datasets Using Educational Data Mining and Machine Learning Models. In *Soft Computing for Intelligent Systems* (pp. 167-175). Springer, Singapore.

[15] Basnet, R. B., Johnson, C., &Doleck, T. (2022). Dropout prediction in Moocs using deep learning and machine learning. *Education and Information Technologies*, 1-15.

[16] Czibula, G., Ciubotariu, G., Maier, M. I., &Lisei, H. (2022). IntelliDaM: A Machine Learning-Based Framework for Enhancing the Performance of Decision-Making Processes. A Case Study for Educational Data Mining. *IEEE Access*, *10*, 80651-80666.

[17] Kaur, H., & Kaur, T. (2023). A Prediction Model for Student Academic Performance Using Machine Learning-Based Analytics. In *Proceedings of the Future Technologies Conference* (pp. 770-775). Springer, Cham.

[18] Raj, N. S., & VG, R. (2022). Early prediction of student engagement in virtual learning environments using machine learning techniques. *E-Learning and Digital Media*, *19*(6), 537-554.

[19] NovilloRangone, G., Pizarro, C., &Montejano, G. (2021, September). Automation of an Educational Data Mining Model Applying Interpretable Machine Learning and Auto Machine Learning. In *International Conference on Communication and Applied Technologies* (pp. 22-30). Springer, Singapore.

[20] Kaur, H., Kaur, T., Bhardwaj, V. et al. An ensemble deep learning model for classification of students as weak and strong learners via multiparametric analysis. *DiscovApplSci 6, 595 (2024)*.

[21] Uskov, V. L., Bakken, J. P., Putta, P., Krishnakumar, D., & Ganapathi, K. S. (2021). Smart education: predictive analytics of student academic performance using machine learning models in Weka and Dataiku systems. In *Smart Education and e-Learning 2021* (pp. 3-17). Springer, Singapore.

[22] Kabathova, J., &Drlik, M. (2021). Towards predicting student's dropout in university courses using different machine learning techniques. *Applied Sciences*, *11*(7), 3130.

[23] Trakunphutthirak, R., & Lee, V. C. (2022). Application of educational data mining approach for student academic performance prediction using progressive temporal data. *Journal of Educational Computing Research*, *60*(3), 742-776.