Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **2**, May **2024**

A Decision Tree and Generalised Regression Neural Network Based Assessment Model for Online Education

Xiang Zhang*

Chongqing Technology and Business Institute,
School of Urban Construction Engineering,
Chongqing, 400052, P. R. China
zhangxiang_1010@163.com

*Corresponding author: Xiang Zhang

Received November 1, 2024, revised January 15, 2024, accepted March 29, 2024.

Abstract. *Online educational assessment models can provide personalised learning recommendations for each student based on their learning data and characteristics. Machine learning techniques can not only model the correlation between academic performance and other factors, but also detect common error patterns of students. Through continuous iteration and training of learning algorithms, the assessment model can predict students' future academic performance. Therefore, an online education assessment model based on decision trees and generalised neural networks is proposed. First, effective indicators for teaching quality assessment are extracted, and the index attributes are quantified and generalised. The entropy gain of each index attribute is calculated according to the samples to be assessed and the entropy gain rate is sorted in descending order. Then, the decision tree model can be used to analyse the learning effect of different student groups and find out the key factors affecting learning. Finally, a Generalised Regression Neural Network (GRNN) is used to train the main variables affecting the prediction of learning performance. After setting the smoothing factor, the learning effect prediction results are obtained after the output of the pattern layer and weighted summation. The experimental results show that a better assessment performance can be obtained by reasonably setting the classification rules and smoothing factors. Compared with the commonly used assessment models, the proposed model can obtain a higher learning effect prediction accuracy.*
Keywords: decision tree; evaluation model; entropy gain; generalised regression neural network; smoothing factor

---

1. Introduction.

Online education assessment models can monitor students' learning in real time. By analysing students' learning data, the assessment model can discover information about students' learning difficulties, learning progress and learning interests in a timely manner [1, 2], provide targeted teaching suggestions and support, and help students learn better.

Online education assessment models can provide teachers with teaching support and feedback. By analysing teachers' teaching data and students' learning data, the assessment model can evaluate the effectiveness and impact of teachers' teaching, provide teachers with suggestions and guidance for teaching improvement, and help teachers to improve their teaching quality and ability. Machine learning can be used to analyse students' learning behaviour data on online learning platforms, such as learning time, number of clicks, page dwell time, etc., from which students' learning patterns and learning habits can be mined to provide a basis for personalised teaching and learning recommendations [3, 4]. Through machine learning algorithms, correlation models can be established between students' academic performance and other factors, such as learning behaviours and the use of learning resources. This can predict students' future academic performance and help teachers and students adjust learning strategies and provide personalised learning support in a timely manner [5, 6]. Machine learning-based assessment models for online education can automatically adjust the learning path and the recommended order of learning resources according to students' learning and personalised characteristics, ensuring that students follow the best learning route and improving learning outcomes and learning satisfaction [7, 8].

Machine learning algorithms can provide customised predictions of academic performance for each student based on their individual characteristics and learning history [9, 10, 11]. Different students have different learning profiles and learning abilities, and machine learning algorithms can take these differences into account to provide predictions that are more relevant to the actual situation of the students. The prediction of academic performance based on machine learning algorithms can be updated in real time to reflect the latest learning status and performance of students. This can provide teachers and students with timely feedback about their academic performance to help them adjust their learning strategies, personalised learning support and improvement [12, 13]. Therefore, the research objective of this work is to classify the key factors affecting the quality of teaching and learning using machine learning algorithms, generate effective classification rules, and obtain the analysis results of each indicator factor. GRNN was then used to train the key variables affecting the prediction of academic performance.

1.1. **Related Work.** machine learning algorithms commonly used in studies related to online education assessment include the following categories.

(1) Linear Regression. Linear regression is a basic regression algorithm that can be used to predict academic performance on a continuous basis. It models the linear relationship between academic performance and other independent variables (e.g., study time, use of learning resources, etc.) Sarker et al. [14] proposed the use of linear regression models to analyse data from large-scale online learning platforms. By modelling the linear relationship between students' study time and academic performance, the effect of study time allocation on academic performance was investigated. The results show that the way of study time allocation has a significant effect on students' academic performance, and the linear regression model can be used to predict students' academic performance.

(2) Decision Tree. A decision tree can be used to predict academic performance by constructing a tree model based on students' learning data. It can consider the relationship between multiple features and generate a series of decision rules to predict academic performance. Martins and Wangenheim [15] constructed a decision tree model using students' behavioural data (e.g., time spent studying, frequency of visits, etc.) and academic performance data for predicting students' academic performance and identifying patterns of learning behaviour. The results of the study showed that the decision tree model was effective in predicting students' academic performance and provided useful suggestions for students' learning behaviours.

(3) Random Forest. Random Forest is an integrated learning method that uses multiple decision trees for learning and prediction. It improves the accuracy of academic performance prediction with good generalisation ability. Tzenios [16] constructed a random forest model using students' academic performance and relevant contextual features (e.g., type of school, students' background, etc.) to predict students' academic performance.

(4) Support Vector Machine (SVM). SVM is a machine learning algorithm for binary classification and regression analysis that can be used for prediction of academic performance. It divides different categories of learning data by finding an optimal hyperplane. Vineetha and Blessie [17] compared four different types of predictive mathematical models including Support Vector Machines (SVMs) for predicting the academic performance of students in an engineering dynamics course.

(5) Neural Networks. Neural networks are algorithms that mimic the workings of neurons in the human brain and can be used for the prediction of academic performance. Lau et al. [18] used a neural network model to build a system that could predict students' final grades. The system can identify factors that may affect academic performance for early intervention and can also be used for test score prediction. The results of the study showed that the neural network model could predict student grades more accurately with an RMSE error of about 8.5 points. The model uses a typical multi-layer feed-forward neural network [19, 20, 21], containing input, hidden and output layers. The input layer contains student's data features, such as attendance hours and homework scores. The hidden layer contains 15 nodes. The output layer has only one node, which is used to predict the student's performance.

1.2. **Motivation and contribution.** Decision trees output rules that are easier to understand and interpret. Random forests consist of multiple decision trees, and their results are not as intuitive as a single decision tree. Therefore, when an interpretable model is required, decision trees are more advantageous. In addition, a single decision tree is prone to overfitting, while Random Forest can reduce overfitting and improve the generalisation ability of the model by integrating predictions from multiple decision trees.

GRNN is able to approximate arbitrary nonlinear functions with good fitting and high accuracy for data pattern recognition. Multi-layer feed-forward neural networks, on the other hand, may encounter the problems of gradient vanishing and local minima [22], which reduce the accuracy. GRNN is insensitive to missing data and outliers through smoothing [23, 24]. And in these cases, the fitting effect of multi-layer feed-forward neural networks is significantly reduced. Therefore, to address the problem of how to improve the accuracy of predicting and assessing learning outcomes, this work proposes an online education assessment model based on decision trees and generalised neural networks.

The main innovations and contributions of this work include:

(1) It is proposed to use C4.5 decision tree to model the relationship between student characteristics (e.g., age, gender, prerequisite course grades, etc.) and learning outcomes (e.g., test scores, course evaluations, etc.). The C4.5 decision tree model can be used to analyse the learning outcomes of different groups of students, and to identify the key factors affecting learning.

(2) Proposes to use GRNN for building learning effect prediction models based on key factors affecting learning (students' video viewing data, homework submission data, forum discussion data, etc.) GRNN can Incremental learning, which can be updated when new data on learning behaviours are available.

(3) Combining the above two, an interpretable decision tree model can be built to analyse the learning effectiveness of different student groups, while the GRNN model can be used to predict the learning effectiveness of individual students in real time. Based on these models, the platform can provide personalised learning suggestions and automatically warn students of possible failure risks.

Compared with traditional teaching assessment, the proposed model can achieve automated, personalised and real-time assessment of teaching quality and learning outcomes. It does not rely on manual statistics, can handle higher dimensional data, and can form a closed loop with the teaching platform to provide assessment feedback to students and teachers, thus contributing to the improvement of teaching quality and learning effectiveness.

2. Decision tree-based relational modelling.

2.1. C4.5 decision tree fundamentals.

C4.5 is a supervised learning algorithm for classification problems [25, 26]. C4.5 decision tree algorithm is an extension of ID3 algorithm. We can create a model to predict the value of the target variable by C4.5 decision tree algorithm and learn the decision rules to complete the classification.C4.5 uses the concepts of Information Gain and Entropy to select the features and classify the dataset. Features with high Information Gain are more likely to be selected as segmentation nodes. Compared to other decision tree algorithms, C4.5 can handle continuous-valued features, whereas ID3 is only suitable for discrete features.C4.5 can handle missing values, whereas ID3 usually just ignores samples with missing values. C4.5 uses bridging techniques to optimise the way continuous values are cut.

The decision tree is mainly composed of root nodes, branch nodes and leaf nodes, and its core structure is shown in Figure 1.

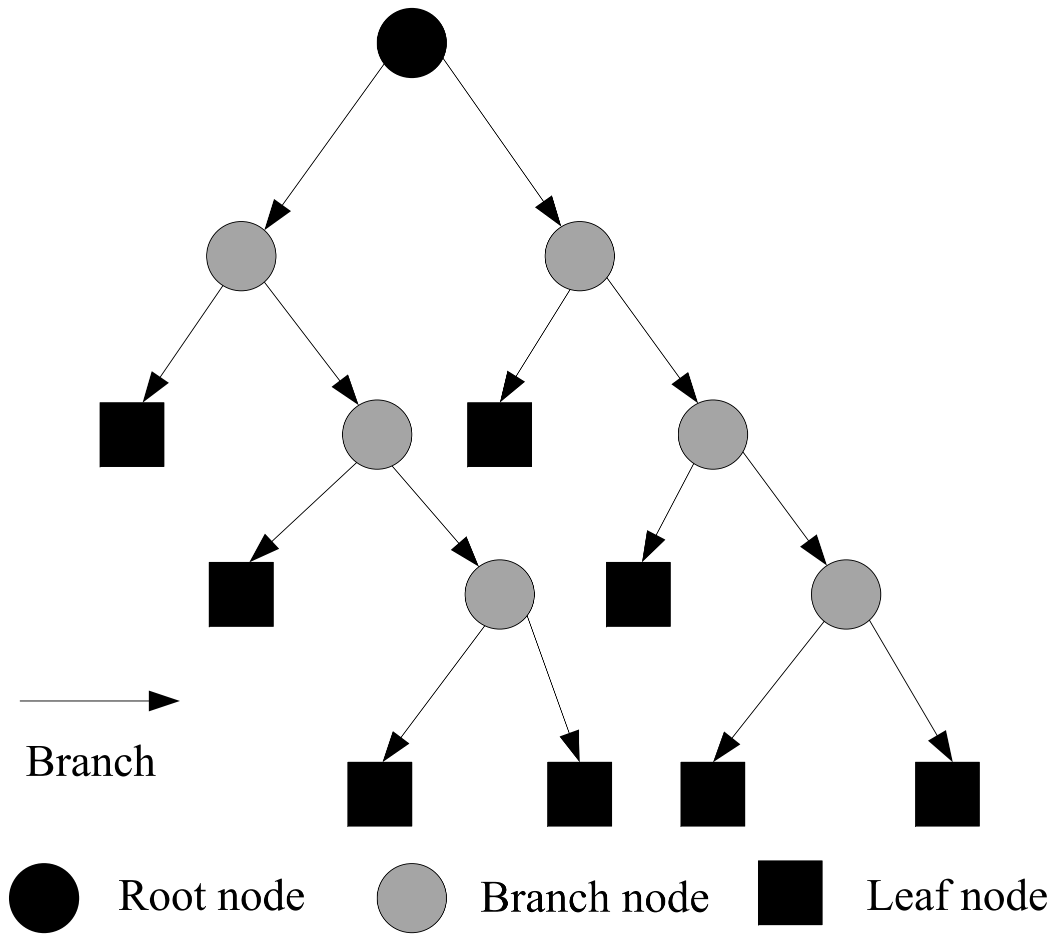

Figure 1. Decision tree core structure

Let the samples in the sample set S be classified into m categories with category C_i; s_i is the number of samples belonging to C_i. The expected entropy of S is calculated as shown below:

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s} \tag{1}$$

where S is the number of samples in the sample set S.

Let some attribute A of the sample be used to subset S. The corresponding expectation E is shown below:

$$E(A) = \sum_{j=1}^{m} \frac{s_{ij} + s_{2j} + \ldots + s_{mj}}{s} I(s_{ij}, s_{2j}, \ldots, s_{mj}) \tag{2}$$

According to Equation (1), the expected entropy of the subset S_i is calculated as shown below:

$$I(s_{ij}, s_{2j}, \ldots, s_{mj}) = -\sum_{i=1}^{m} \frac{s_{ij}}{s_j} \log_2 \frac{s_{ij}}{s_j} \tag{3}$$

The expected entropy gain of A over S is calculated as shown below:

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A) \tag{4}$$

If expressed in terms of gain rate, the expected entropy gain is calculated as shown below:

$$\text{Gain}(A) = \frac{\text{Gain}(A)}{\text{splitInfo}(s)} \tag{5}$$

where

$$splitInfo(s) = \sum_{i=1}^{m} \frac{s_i}{|s|} \times \log_2 \frac{s_i}{|s|} \tag{6}$$

In the C4.5 decision tree algorithm, the information gain ratio is used as a substitute for the information gain degree to select the optimal features and avoid bias towards multi-valued features. The information gain ratio is calculated as follows.

$$\text{IGR}(D, A) = \frac{\text{IG}(D, A)}{\text{IV}(A)} \tag{7}$$

where $\text{IG}(D, A)$ is the information gain of feature A with respect to dataset D and $\text{IV}(A)$ is the Impurity of feature A.

Information gain indicates the increase in the amount of information that can be brought about by selecting features [27].

$$\text{IG}(D, A) = H(D) - H(D|A) \tag{8}$$

where $H(D)$ is the entropy of the dataset D and $H(D|A)$ is the conditional probability entropy after partitioning according to A. The conditional probability entropy $H(D|A)$ is used to measure the uncertainty of each separated set after selecting features.

$$H(D|A) = \sum_{v \in \text{Values}(A)} \left(\frac{|D_v|}{|D|} \right) * H(D_v) \tag{9}$$

where D_v denotes the subset divided according to the value v of feature A.

The basic computational principle of C4.5 algorithm is given above, which uses information theory to select the best features and construct a decision tree. In the actual implementation, we need to consider pruning and other optimisation strategies.

2.2. **Modelling the relationship between student characteristics and learning outcomes.** In this work, C4.5 decision tree is used to model the relationship between student characteristics (e.g. age, gender, prerequisite grades, etc.) and learning outcomes (e.g. test scores, course evaluations, etc.), so as to analyse the learning outcomes of different groups of students, and to find out the key factors affecting learning.

A C4.5 decision tree is a binary tree in which each node represents a feature and each branch represents a possible feature value. The root node of the tree represents the entire dataset while the leaf nodes of the tree represent each class in the dataset.The training process of C4.5 decision tree is as follows:

(1) Select a feature from the training set that is the feature in the dataset that best distinguishes the different classes.

(2) Divide the dataset into two subsets, one subset containing instances with that feature value and the other subset containing instances without that feature value.

(3) Repeat steps 1 and 2 for each subset until each subset contains only one class.

The forecasting process for the C4.5 decision tree is as follows:

(1) Starting from the root node of the tree, a branch is selected based on the eigenvalues of the instance.

(2) Continue along the path of the tree until you reach a leaf node.

(3) Leaf nodes indicate the class to which the instance belongs.

The specific steps for analysing student learning outcomes using the C4.5 decision tree are shown below:

Step 1: Data preparation is needed. Collect more representative data on students, including their demographic characteristics (age, gender, etc.), family background (parental education, family income, etc.), learning characteristics (high school grades, extracurricular activities, etc.), and learning outcomes (test scores, dropout rates, course evaluations, etc.).

Step 2: Check and process missing values, convert categorical features to numerical features, normalise numerical features, etc.

Step 3: Split the dataset into training dataset and validation dataset in 7:3 ratio.

Step 4: Train the C4.5 decision tree model on the training set. Construct a classification tree using the C4.5 algorithm. Calculate the information gain ratio for each feature and select the feature with the largest information gain ratio as the node feature [28]. For each value of the node feature, the steps are repeated recursively for a subset of the data, recursively until the termination condition is satisfied. Termination conditions include the subset of samples all belonging to one class, no more features, or reaching a preset depth.

Step 5: For each feature A, calculate the information gain ratio $IGR\ (D,\ A)$ for A.

Step 6: Select the feature with the largest information gain ratio as the segmentation node [29, 30].

Step 7: Recursively repeat Steps 4-6 for the partitioned subset until the stop condition is satisfied.

Step 8: Test the model using the test set.

The specific method for constructing the learning effect relationship model based on the C4.5 decision tree is shown in Algorithm 1.

Algorithm 1 Constructing learning effect relationship model based on C4.5 decision tree

Input: Training set D (feature set X, label set Y)
Output: Decision tree T

Function C4.5Decision_Tree(D, feature set X).
if all instances belong to the same class C **then**
  return a single-node tree Root, labelled C
end if
if feature set is empty **then**
  return a single-node tree Root, labelled with majority class in D
else
  Select the best feature A to split on, split D into subsets D_i
  for each subset D_i **do**
    Subtree $T_i = \text{C4.5}(D_i, X - A)$
  end for
  return a tree with root A and branches T_i
end if

3. GRNN-based model for predicting learning outcomes.

3.1. Feature extraction of key factors affecting learning.

This work proposes to use GRNN for building a prediction model of learning effectiveness based on students' video viewing data, homework submission data, and forum discussion data. Therefore, firstly, features of key factors affecting learning need to be extracted.

(1) Video viewing data. The number of videos watched is calculated as shown below:

$$V_{num} = \Sigma V_i \tag{10}$$

where V_i denotes the number of times the i-th video was viewed

The average viewing time is calculated as shown below:

$$V_{avg_{dur}} = \frac{\sum V_i \cdot Dur_i}{V_{\text{num}}} \tag{11}$$

where Dur_i denotes the length of the i-th video.

The viewing frequency is calculated as shown below:

$$V_{\text{freq}} = \frac{\sum V_w / T}{W} \tag{12}$$

where V_w denotes the number of views in a week, T denotes the total time in a week, and W denotes the total number of weeks.

The repeat view rate is calculated as shown below:

$$V_{\text{rerate}} = \frac{\sum V_{i,\text{r}}}{V_{\text{num}}} \tag{13}$$

where $V_{i,\text{r}}$ denotes the number of repeated viewings of the i-th video.

Job submission data. The on-time submission rate is calculated as shown below:

$$H_{ontime_{rate}} = \frac{H_{\text{ontime}}}{H_{\text{total}}} \tag{14}$$

where H_{ontime} denotes the number of on-time submissions and H_{total} denotes the total number of job submissions.

The number of job revisions is calculated as shown below:

$$H_{\text{modify}} = \frac{\sum H_{i,m}}{H_{\text{total}}} \tag{15}$$

where $H_{i,m}$ denotes the number of revisions per job i.

The high mark rate for assignments is calculated as shown below:

$$H_{high_{rate}} = \frac{H_{\text{high}}}{H_{\text{total}}} \tag{16}$$

where H_{high} indicates the number of assignments that received a high score.

The assignment score rate is calculated as shown below:

$$H_{score_{rate}} = \frac{\sum H_{i,s}}{\sum H_{i,\max}} \tag{17}$$

where $H_{i,s}$ and $H_{i,\max}$ are the score and full score of assignment i, respectively.

Forum discussion data. The number of posts is calculated as shown below:

$$P_{\text{num}} = \sum P_i \tag{18}$$

where P_i denotes the i-th original post.

The posting word count is calculated as shown below:

$$P_{\text{words}} = \frac{\sum P_{i,w}}{P_{\text{num}}} \tag{19}$$

where $P_{i,w}$ is the word count of post i. The number of participants in the discussion is calculated as shown below:

$$D_{\text{num}} = \sum D_i \tag{20}$$

where D_i denotes the i-th discussion involved.

The number of endorsements received is calculated as shown below:

$$P_{\text{endors}} = \sum P_{i,e} \tag{21}$$

where $P_{i,e}$ is the number of endorsements received by post i. The approval rate is calculated as shown below:

$$P_{endorse_{rate}} = \frac{\sum P_{i,e}}{P_{\text{num}}} \tag{22}$$

The features extracted from these equations can effectively reflect the students' learning input and establish the learning effect prediction model. These 13 features can reflect students' learning effects from different dimensions such as learning time investment, learning attitude and initiative, learning quality, etc. They are representative and distinguishable, so they are chosen as feature parameters. These parameters are highly correlated with the learning effects and are suitable for building prediction models.

3.2. **Construction of the prediction model.** GRNN is a neural network based regression algorithm, the core idea of which is to use Gaussian function and distance weights to regress the input features for prediction. This work uses the key factors affecting learning to construct a GRNN-based prediction model for learning effectiveness. The input layer contains 13 data characteristics of students, such as the number of videos watched and the number of homework revisions. The hidden layer contains 15 nodes. There are six nodes in the output layer, which are used to predict students' scores in six courses.

In the GRNN model set the number of input cells as $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}^T$, the pattern layer contains n cells and the pattern layer transfer function is shown below:

$$p_i = \exp[-\frac{(\mathbf{X} - \mathbf{X_i})^T(\mathbf{X} - \mathbf{X_i})}{2\sigma^2}] \tag{23}$$

where $\mathbf{X}$ represents the input variable, $\mathbf{X}_i$ represents the i-th training sample, P_i is the pattern layer output, and σ represents the smoothing factor.

The structure of the GRNN model is shown in Figure 2.

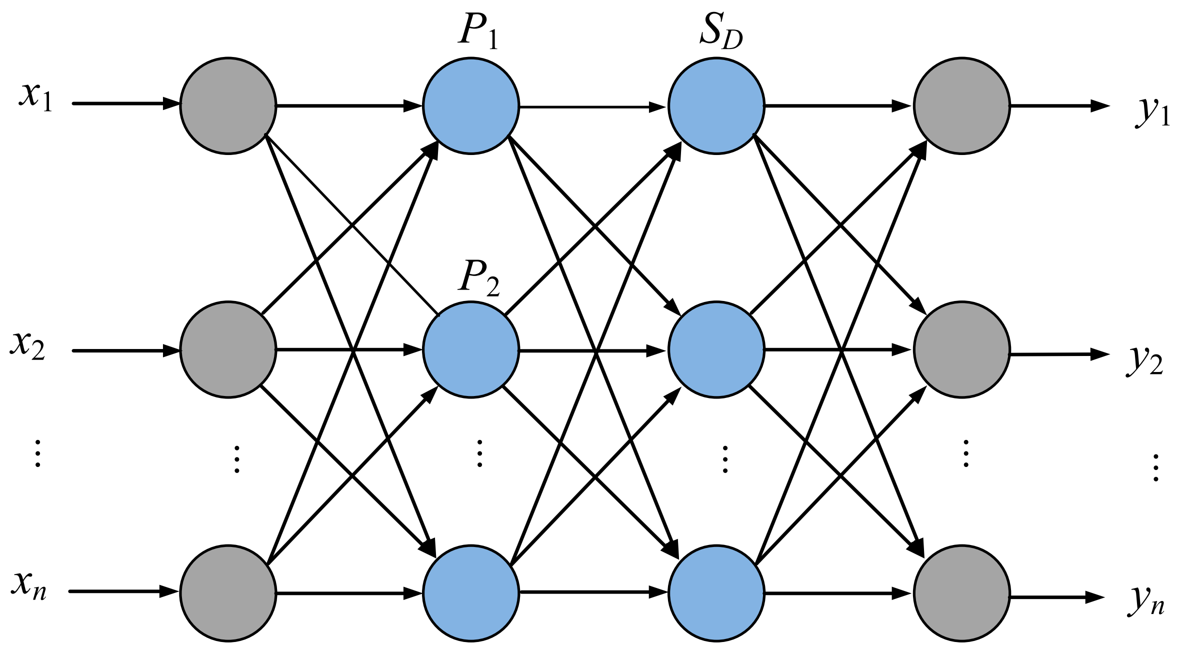

Figure 2. Structure of the GRNN model

It is often necessary to perform a summation operation on the output of the GRNN mode layer.

$$S_D = \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} \exp[-\frac{(\mathbf{X}-\mathbf{X_i})^T(\mathbf{X}-\mathbf{X_i})}{2\sigma^2}] \tag{24}$$

The following equation is obtained after weighted summation.

$$S_{Nj} = \sum_{i=1}^{n} y_{ij} \exp[-\frac{(\mathbf{X}-\mathbf{X_i})^T(\mathbf{X}-\mathbf{X_i})}{2\sigma^2}], j = 1, 2, \dots, k \tag{25}$$

where y_{ij} denotes the j-th element of the i-th output sample. The results of the output layer are shown below:

$$y_j = \frac{S_{Nj}}{S_D} \tag{26}$$

where k denotes the output vector dimension.

In the proposed GRNN based learning effect prediction model, the input feature vector is defined as $X = [x_1, x_2, \dots, x_m]$, where x_i denotes the i-th feature, such as 13 features like repeat viewing rate, number of assignment revisions, and number of posts. The output vector is defined as $Y = [y_1, y_2, \dots, y_n]$, where y_i denotes the learning effect of the i-th student. Therefore, the core equation of the proposed learning effect prediction model is shown below:

$$f(x) = \frac{\sum\limits_{i=1}^{N} \exp(-\frac{||x-x_i||^2}{2\sigma^2}) y_i}{\sum\limits_{i=1}^{N} \exp(-\frac{||x-x_i||^2}{2\sigma^2})} \tag{27}$$

where $\mathbf{x}$ is the input feature vector, x_i is the feature vector in the training set, y_i is the learning effect data (target value) corresponding to x_i, and σ is the parameter (smoothing factor) that controls the kernel width. We can choose the appropriate value of σ by cross-validation.

The GRNN neural network model, constructed in this work, consists of an input layer, a template layer, an association layer and an output layer.

(1) Template layer training: the template layer is trained using feature data and learning effect data. The template layer is the core part of GRNN, which is used to store the features and corresponding learning effects of the training samples. Each template vector t_i of the template layer consists of feature data x_i and corresponding learning effect y_i.

$$t_i = [x_i, y_i] \tag{28}$$

(2) Correlation layer computation: the correlation between each template and the input is computed based on the input features and samples from the template layer. The correlation between the input feature x_i and the template vector t_i is calculated using the Gaussian kernel function.

$$R_i = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \tag{29}$$

where $\|x - x_i\|^2$ denotes the Euclidean distance between the input features and the template features.

(3) Output layer calculation: based on the correlation and the corresponding learning effect, the final learning effect prediction result is calculated.

$$\text{Prediction} = \frac{\sum(R_i \cdot y_i)}{\sum R_i} \tag{30}$$

where R_i is the degree of association and y_i is the corresponding learning effect.

The pseudo-code of the GRNN-based learning effect prediction model is shown in Algorithm 2.

In addition, the performance of GRNN is also affected by the kernel width parameter (σ), so the appropriate σ needs to be chosen to obtain the best performance.

4. **Experimental results and analyses.**

4.1. **Experimental environment and experimental dataset.**

The proposed model was simulated and tested using Python programming language and PyTorch software in Windows 7 operating system. In order to validate the performance of C4.5 Decision Tree and GRNN in teaching quality assessment, a Massive Open Online Course (MOOC) platform was used for training and testing.MOOCs sometimes release datasets on student behaviour and academic performance. This dataset contains student performance data from a massive open online course (MOOC).

MOOC datasets include features such as students' online behaviour, quiz scores, forum activity, and labels for course completion. These data typically include a large amount of student information that can be used for educational data mining and performance evaluation. Therefore, this work uses this dataset to construct models to predict student performance in a MOOC. Firstly, decision tree classification is performed on the assessment samples to obtain teaching quality classification rules, and then GRNN is used to predict learning outcomes.

4.2. **Classification effect of decision tree.**

The teaching quality data to be assessed are generated data records one by one, and then the entropy gain of the nine indicators is calculated according to the relational model based on C4.5 decision tree, and its statistical results are shown in Table 1.

According to the *Gain(A)* value in Table 1, the title attribute has a maximum value of 0.071, so the title attribute is the root node of the decision tree. Generate 4 different branches based on the 4 attribute values of the job title and then select the teaching

Algorithm 2 GRNN-based Learning Effect Prediction Model

```
#Template layer training
def template_layer_training(features, labels)
templates ← []
for i ← 1 to length(features) do
    template ← [features[i], labels[i]]
    templates.append(template)
end for
return templates
# Associative layer calculations
def compute_relation(input_feature, templates, σ)
relations ← []
for template in templates do
    distance ← calculate_distance(input_feature, template[0])
    relation ← exp(−distance^2/(2σ^2))
    relations.append(relation)
end for
return relations
# Output layer calculations
def compute_output(relations, templates)
numerator ← 0.0
denominator ← 0.0
for i ← 1 to length(relations) do
    numerator ← numerator + relations[i] × templates[i][1]
    denominator ← denominator + relations[i]
end for
return numerator/denominator
# Model training
def train_model(features, labels, sigma)
templates ← template_layer_training(features, labels)
return templates, sigma
# Model predictions
def predict(features, templates, sigma)
predictions ← []
for feature in features do
    relations ← compute_relation(feature, templates, sigma)
    prediction ← compute_output(relations, templates)
    predictions.append(prediction)
end for
return predictions
# Data preparation
features_train ← [...] {features for training}
labels_train ← [...] {labels for training}
features_test ← [...] {features for testing}
# Model training
templates, σ ← train_model(features_train, labels_train, sigma)
# Model predictions
predictions ← []
for feature in features_test do
    prediction ← predict(feature, templates, sigma)
    predictions.append(prediction)
end for
# Output prediction results
print(predictions)
```

Table 1. Entropy gain of evaluation index.

Indicator properties	E(A)	Gain(A)
X	1.073	0.063
Z	1.078	0.068
N	1.046	0.023
T	1.074	0.065
P	0.999	0.003
S	1.005	0.006
D	1.009	0.008
L	1.058	0.038
R	1.002	0.004

attitude as the branch node to continue building the decision tree. Continue to select branch nodes based on *Gain(A)* values to keep expanding the decision tree until all *Gain(A)* values nodes have been calculated. Finally the IF-THEN classification rules obtained based on the tree structure. One of the classification rules rated as excellent is shown in Table 2.

Table 2. Classification rules

Scoring Excellence Rules
IF (Title = Associate AND Education = Master's) THEN (Teaching Rating Excellent)
IF (Title = Associate AND Teaching Attitude = Good) THEN (Teaching Rating Excellent)
IF (title = full senior AND teaching attitude = good) THEN (teaching rating excellent)
IF (Title = Intermediate AND Teaching Attitude = Good) THEN (Teaching Rating Excellent)
IF (Education = Masters AND Teaching Attitude = Good) THEN (Teaching Rating Excellent)

4.3. **Prediction performance of GRNN for different σ values.** In the following, 15 variables of five indicators, x_1, x_2, x_3, x_4, and x_5, are used for GRNN training. According to the experience of previous GRNN model research, the value of σ is generally selected in the range of [0.6, 0.8]. The learning effect of 30 samples was predicted by GRNN, and the relative error mean of 30 samples was counted. The relative error of prediction for different σ values is shown in Figure 3.

It can be seen that the learning effect prediction error decreases firstly and then rises with the increase of σ value. When $\sigma = 0.65$, the average error of prediction error is about 0.04, and when σ is greater than 0.65, the prediction error climbs rapidly. So, in this paper, we choose $\sigma = 0.65$ to be suitable for the sample training task of this work.

4.4. **Performance analysis of prediction accuracy.** The GRNN algorithm chooses $\sigma = 0.65$. The minimum number of split samples (Min Split) for the C4.5 decision tree is set to 20 and the minimum number of leaf samples (Min Leaf) is set to 10 to ensure that there are enough samples in each leaf node. The Split Criterion was chosen to use Gini Impurity. The Max Depth of the tree was set to 5 to limit the depth of the tree to avoid overfitting. The tree is trained on 30 training samples, then the learning effect of 6 courses is predicted and finally compared with the actual learning effect. The predicted results of academic achievement are shown in Figure 4.

It can be seen that the predicted academic performance obtained using the C4.5+GRNN algorithm is very close to the actual academic performance, especially for two courses, English course and Physical Education (PE) course, where the predicted and actual values

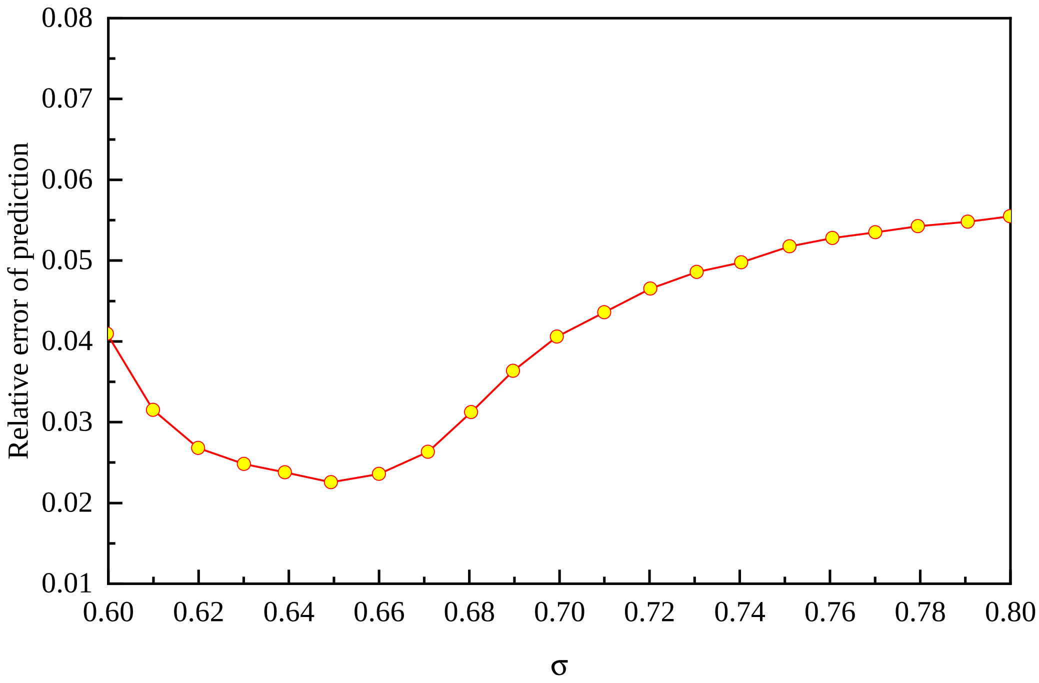

Figure 3. Relative error of prediction for different σ values

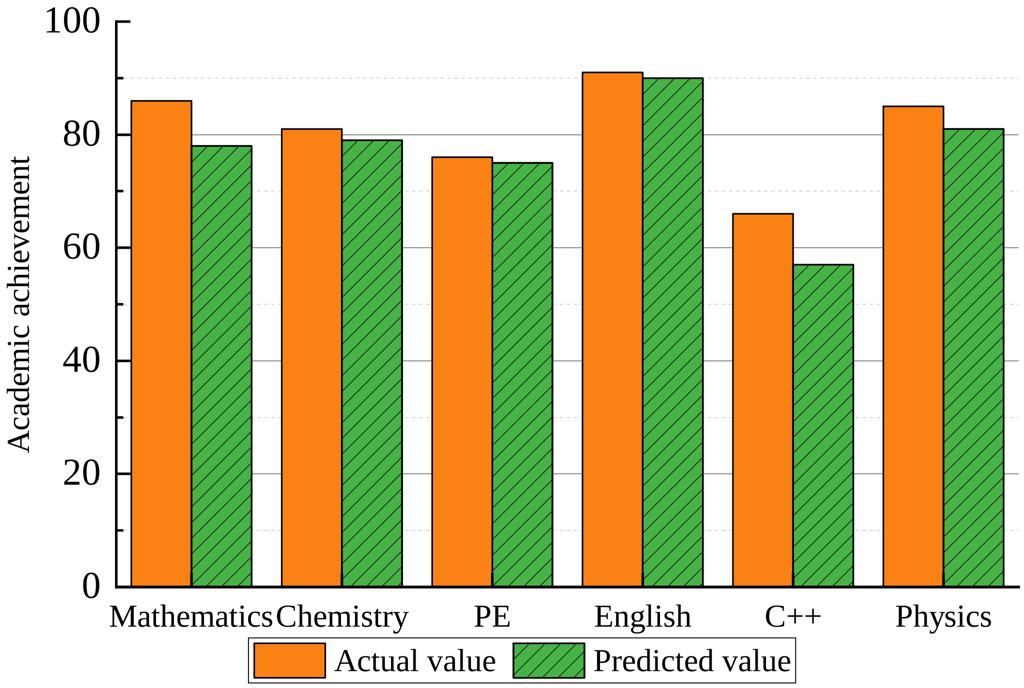

Figure 4. Predicted results of academic achievement

are very close to each other. While C++ and Maths showed some prediction bias. And for all the courses, the predicted academic performance is smaller than the actual academic performance. The relative error of prediction is shown in Figure 5.

In terms of the relative errors of prediction, the errors of prediction of academic performance of the six courses are all within the range of 0.04, among which the English course has the best prediction result with an error value of nearly 0.006, and the PE course is

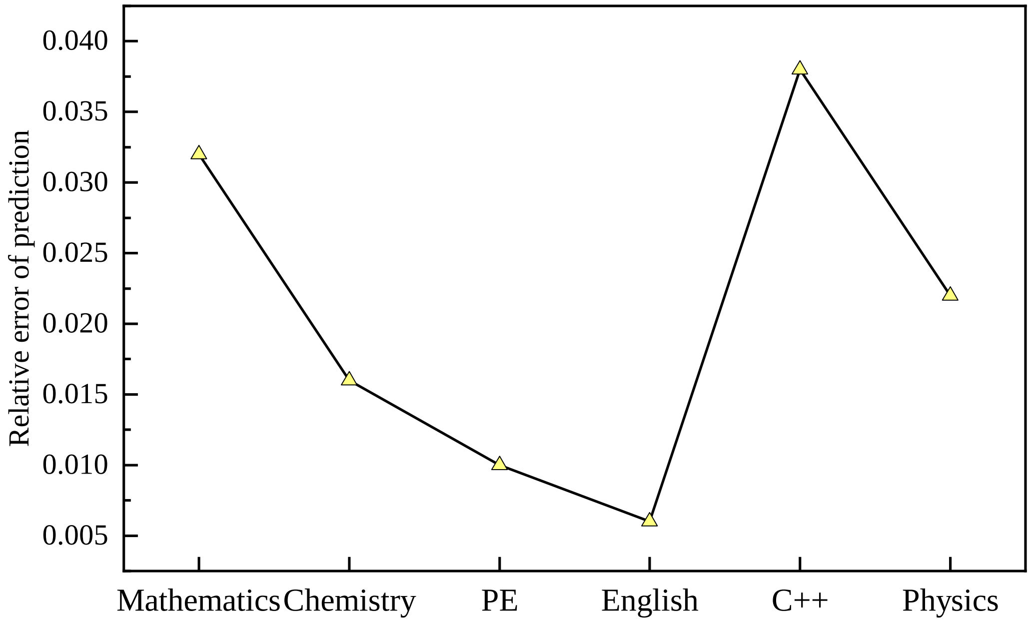

Figure 5. Relative error of prediction

the second best with an error of 0.01. The C++ course has the worst prediction result, with an error of about 0.038, and the average relative error of the six courses is about 0.02.

4.5. **Prediction performance of different algorithms.** In order to further validate the performance of C4.5+GRNN model in the prediction of learning outcomes, comparative analyses were conducted with Random Forest [16], SVM [17] and Multilayer Feedforward Neural Network [18]. All the four algorithms predicted the learning outcomes of six courses. The relative error of prediction with three different algorithms is shown in Figure 6.

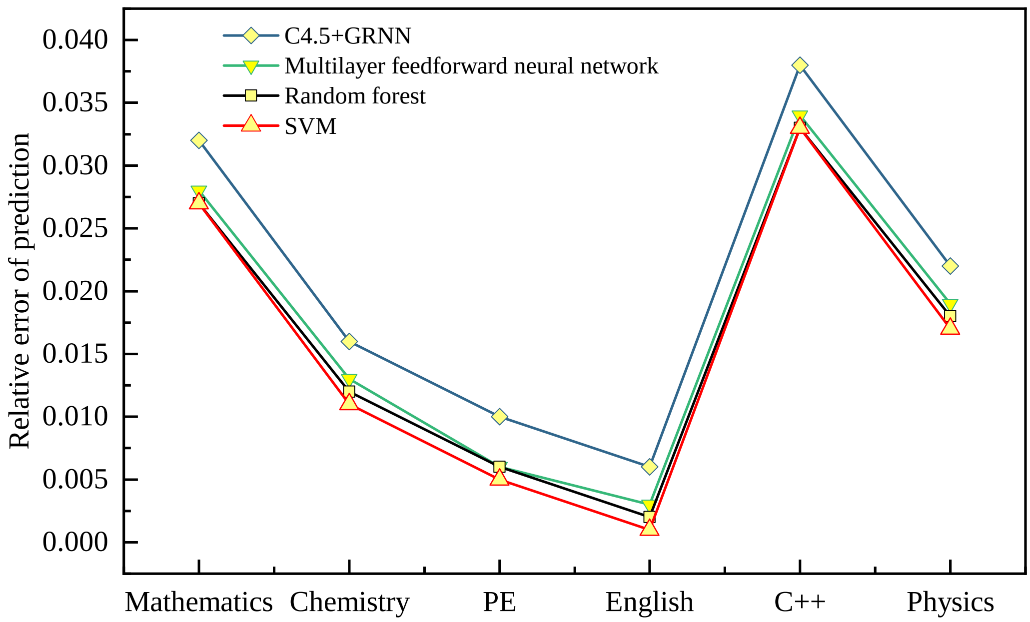

Figure 6. Relative error of prediction with three different algorithms

It can be seen that all four algorithms have some effect on the fitting of the actual learning effect, among which the learning effect prediction curve of C4.5+GRNN model is the closest to the actual learning effect curve, and the learning effect curve of SVM is the worst fitted. For the C++ course, the learning effect prediction performance of the four algorithms is significantly reduced, probably because the selection of key factors affecting the learning effect is not in line with the learning pattern of the C++ course. While the indicators selected in this paper mainly analyse the main factors affecting the learning effect from the aspect of macro factors, later research will incorporate more consideration indicators to improve the accuracy of the learning effect prediction analysis.

5. **Conclusion.** To address the problem of how to improve the accuracy of predicting and assessing learning outcomes, this work proposes an online education assessment model based on C4.5 decision trees and generalised neural networks. First, it is proposed to use decision trees to model the relationship between student characteristics and learning outcomes. The decision tree model can analyse the learning effectiveness of different student groups and identify the key factors affecting learning. Secondly, GRNN is proposed to be used to build a prediction model of learning effectiveness based on the key factors affecting learning. By combining the above two, an interpretable decision tree model can be built to analyse the learning effects of different student groups, and at the same time, the GRNN model can be used to predict the learning effects of individual students in real time. The experimental results show that the C4.5+GRNN model is able to obtain a better assessment performance by reasonably setting the classification rules and smoothing factors. However, since the GRNN model is used in this paper, there are no model parameters to be set, but for the test samples all the training samples are involved in the calculation, i.e., each test sample has to be calculated with all the training samples, so the computational complexity is high. Further research will be carried out on how to reduce the computational complexity.

REFERENCES

[1] J. W. Gikandi, D. Morrow, and N. E. Davis, "Online formative assessment in higher education: A review of the literature," *Computers & Education*, vol. 57, no. 4, pp. 2333-2351, 2011.

[2] A. Joshi, M. Vinay, and P. Bhaskar, "Impact of coronavirus pandemic on the Indian education sector: perspectives of teachers on online teaching and assessments," *Interactive Technology and Smart Education*, vol. 18, no. 2, pp. 205-226, 2021.

[3] S. McGrew, J. Breakstone, T. Ortega, M. Smith, and S. Wineburg, "Can students evaluate online sources? Learning from assessments of civic online reasoning," *Theory & Research in Social Education*, vol. 46, no. 2, pp. 165-193, 2018.

[4] F. Martin, A. Ritzhaupt, S. Kumar, and K. Budhrani, "Award-winning faculty online teaching practices: Course design, assessment and evaluation, and facilitation," *The Internet and Higher Education*, vol. 42, pp. 34-43, 2019.

[5] A. D. Dumford, and A. L. Miller, "Online learning in higher education: exploring advantages and disadvantages for engagement," *Journal of Computing In Higher Education*, vol. 30, pp. 452-465, 2018.

[6] M. Lynch, T. Sage, L. I. Hitchcock, and M. Sage, "A heutagogical approach for the assessment of Internet Communication Technology (ICT) assignments in higher education," *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, 55, 2021.

[7] A. Selvaraj, V. Radhin, K. Nithin, N. Benson, and A. J. Mathew, "Effect of pandemic based online education on teaching and learning system," *International Journal of Educational Development*, vol. 85, 102444, 2021.

[8] J. Maldonado-Mahauad, M. Pérez-Sanagustín, R. F. Kizilcec, N. Morales, and J. Munoz-Gama, "Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses," *Computers in Human Behavior*, vol. 80, pp. 179-196, 2018.

[9] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, 40, 2019.

[10] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[11] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, and T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17-29, 2019.

[12] C.-M. Chen, Z. Zhang, J. Ming-Tai Wu, and K. Lakshmanna, "High Utility Periodic Frequent Pattern Mining in Multiple Sequences," *Computer Modeling in Engineering & Sciences*, vol. 137, no. 1, pp. 733-759, 2023.

[13] C.-M. Chen, Y. Gong, and J. M.-T. Wu, "Impact of Technical Indicators and Leading Indicators on Stock Trends on the Internet of Things," *Wireless Communications and Mobile Computing*, vol. 2022, 9916310, 2022.

[14] I. H. Sarker, A. Kayes, and P. Watters, "Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage," *Journal of Big Data*, vol. 6, no. 1, pp. 1-28, 2019.

[15] R. M. Martins, and C. G. V. Wangenheim, "Findings on teaching machine learning in high school: A ten-year systematic literature review," *Informatics in Education*, vol. 22, no. 3, pp. 421-440, 2023.

[16] N. Tzenios, "Examining the Impact of EdTech Integration on Academic Performance Using Random Forest Regression," *ResearchBerg Review of Science and Technology*, vol. 3, no. 1, pp. 94-106, 2020.

[17] K. R. Vineetha, and E. C. Blessie, "Efficient prediction of Student Performance Using hybrid SVM Classifier," *International Journal of Computer Science & Engineering Technology*, vol. 9, no. 3, pp. 32-39, 2018.

[18] E. Lau, L. Sun, and Q. Yang, "Modelling, prediction and classification of student academic performance using artificial neural networks," *SN Applied Sciences*, vol. 1, pp. 1-10, 2019.

[19] M. A. Rubi, S. Chowdhury, A. A. A. Rahman, A. Meero, N. M. Zayed, and K. A. Islam, "Fitting multi-layer feed forward neural network and autoregressive integrated moving average for Dhaka Stock Exchange price predicting," *Emerging Science Journal*, vol. 6, no. 5, pp. 1046-1061, 2022.

[20] P. N. Suganthan, and R. Katuwal, "On the origins of randomization-based feedforward neural networks," *Applied Soft Computing*, vol. 105, 107239, 2021.

[21] N. J. Guliyev, and V. E. Ismailov, "On the approximation by single hidden layer feedforward neural networks with fixed weights," *Neural Networks*, vol. 98, pp. 296-304, 2018.

[22] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks," *Neural Computing and Applications*, vol. 29, pp. 991-1004, 2018.

[23] I. Izonin, N. Kryvinska, R. Tkachenko, K. Zub, and P. Vitynskyi, "An extended-input GRNN and its application," *Procedia Computer Science*, vol. 160, pp. 578-583, 2019.

[24] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Surface EMG hand gesture recognition system based on PCA and GRNN," *Neural Computing and Applications*, vol. 32, pp. 6343-6351, 2020.

[25] X. Zheng, W. Feng, M. Huang, and S. Feng, "Optimization of PBFT algorithm based on improved C4. 5," *Mathematical Problems in Engineering*, vol. 2021, pp. 1-7, 2021.

[26] T. H. Sinaga, A. Wanto, I. Gunawan, S. Sumarno, and Z. M. Nasution, "Implementation of Data Mining Using C4. 5 Algorithm on Customer Satisfaction in Tirta Lihou PDAM," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 3, no. 1, pp. 9-20, 2021.

[27] P. V. Ngoc, C. V. T. Ngoc, T. V. T. Ngoc, and D. N. Duy, "A C4. 5 algorithm for english emotional classification," *Evolving Systems*, vol. 10, no. 3, pp. 425-451, 2019.

[28] M. M. Mijwil, and R. A. Abttan, "Utilizing the genetic algorithm to pruning the C4. 5 decision tree algorithm," *Asian Journal of Applied Sciences*, vol. 9, no. 1, pp. 514-521, 2021.

[29] X. Meng, P. Zhang, Y. Xu, and H. Xie, "Construction of decision tree based on C4. 5 algorithm for online voltage stability assessment," *International Journal of Electrical Power & Energy Systems*, vol. 118, 105793, 2020.

[30] B. Konakoglu, "Prediction of geodetic point velocity using MLPNN, GRNN, and RBFNN models: A comparative study," *Acta Geodaetica et Geophysica*, vol. 56, no. 2, pp. 271-291, 2021.