KeAi
CHINESE ROOTS GLOBAL IMPACT

International Journal of Innovation Studies

ISSN:2096-2487 | E-ISSN:2589-2975

KERNEL DISCRIMINANT OPTIMIZED DEEP CONVOLUTIONAL NEURAL LEARNING CLASSIFIER FOR PREDICTIVE ANALYTICS

P. Christy Jeyakumar[a*], Dr. S. Ismail Kalilulah[b], Dr. V. N. Rajavarman[c]

[a] Ph.D. Research Scholar, Department of Computer Applications, Dr. M.G.R. Educational and Research Institute, Chennai

[b] Professor, Department of Computer Science and Engineering, Dr. M.G.R. Educational and Research Institute, Chennai

[c] Professor, Department of Computer Science and Engineering, Dr. M.G.R. Educational and Research Institute, Chennai

* Corresponding Author Email: p.christyjeyakumar@gmail.com

Abstract

Data mining (DM) is an emerging field that focuses on applying big datasets of mining techniques to enhance the results and learning process. It involves find out the patterns and correlations within huge sets of data with the aim of forecasting future outcomes. Predicting future outcomes for large datasets is a critical application of data mining. By analyzing historical admission data and other relevant factors, existing predictive models have been developed to forecast the likelihood of various outcomes. However, these models faced challenges when applied to large datasets, affecting their predictive effectiveness. To improve prediction accuracy, a novel model called Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier (KDODCNLC) is introduced. The primary aim of the KDODCNLC model is to achieve higher prediction accuracy and reduced time consumption. In this model, data is first collected during the data acquisition phase for classification purposes. The proposed DCNN comprises the different layers. At first, data is fed into input layer from Student Performance Prediction dataset. The input data then moves to hidden layer 1, where preprocessing is performed to remove unwanted data. The preprocessed data is passed to hidden layer 2. In this max-pooling layer, feature selection is conducted to identify significant features while discarding others. Then, it is sent to a fully connected layer for classification to reduce the features. In this layer, data is determined by a kernelized discriminant function to provide prediction outcomes. To minimize classification error, hyperparameter optimization in Convolutional Neural Networks (CNNs) is essential, involving fine-tuning of network parameters. The KDODCNLC model employs the Rain-fall optimization algorithm for hyperparameter optimization, aiming to reduce error and enhance prediction accuracy. To conclude, the accurate prediction outcomes are attained at output layer, enabling predictions through less time consumption. Experimental evaluation obtains the different metrics. Outcome quantitatively demonstrates KDODCNLC model enhances the prediction accuracy by 5%, precision by 5%, and recall by 3% and F1-Scoreby 4%, then the conventional methods. Additionally, KDODCNLC achieves a 27% reduction in prediction time than the existing methods.

***Keywords**: Data mining, prediction, deep convolutional neural network, kernelized discriminant function, hyperparameter optimization, rain-fall optimization algorithm*

1. Introduction

Predictive data mining techniques have many uses in data mining and are used to forecast future events using historical data. This process is essential across various domains, including business, finance, healthcare, transportation, education, industry, e-commerce, and telecommunications. Prediction in data mining plays a significant role in these sectors by providing insights that enhance learning experiences, improve outcomes, and optimize operations. In recent days, accurately measuring student performance is a challenging issue using significant features. The designed ML and DL methods are focused to be increasing algorithms and to study from data and make predictions or decisions.

Feature selection is crucial for predicting student performance. VR-ODRNN was introduced in [1] for selecting relevant features with less time. However, the student's quality of learning was not concentrated. Also, the huge dataset was not concentrated. To address this issue, the students' performance was predicted in [2] by using PSO-DNN. Designed PSO-DNN was employed to determine near-optimal solutions for enhancing the accuracy and student learning quality. However, it failed to focus on the admissions system.

An admissions system is significant for performing decision-making to predict applicants' academic performance at university. The ANN was designed in [3] to improve prediction performance compared to other classification methods. This method provides an early prediction of university performance before admission based on specific criteria. However, it did not perform multiclass categorization.

To enhance accuracy, a multiclass student result prediction model (RLCHI) was introduced in [4]. However, the missing data was not handled. Study focused on hybrid CNN and LSTM in [5] for reducing time. In addition, the designed CNN-LSRM handles missing information as well as the information imbalance issue. Nevertheless, accuracy was not enhanced.

A DL model was presented [6] to enhance accurately predict student performance. But, it did not recognize the needs of students at risk of academic failure using classification. Yet another deep learning model was investigated in [7] by employing Bi-LSTM network. The best classifiers were determined to forecast the at-risk students. However, the dimensionality was not reduced.

Students performance is vital for the learning process. Machine learning algorithms were designed in [8] to improve student performance prediction through dimensionality reduction. However, these algorithms failed to enhance prediction performance with minimal complexity. Yet another machine learning method using pattern discovery techniques was developed in [9]. However, the complexity level of the performance was not reduced. In [10], a NN as well as Naïve Bayes techniques were employed into predict academic performance based on student features. However, the dynamic neural network model was not predicted or classified the student performance.

The students' grade point average (GPA) based on their profiles is predicted by using method of K-means clustering algorithm developed in [11]. However, it failed to select the most relevant variables for the student admission process, given the criteria used by educational institutions. In [12], the data mining pattern discovery methods were developed through feature selection for predicting models for university admissions and graduation performance. But, it failed to improve accuracy of forecast.

With intend of improving predictive methods for graduate admissions based on candidate parameters, Naïve Bayes as well as Logistic Regression classification methods were designed [13]. However, these models did not effectively utilize machine learning methods to enhance predictive capabilities in various fields, including medical diagnosis, financial trading, and educational performance. Yet another Naïve Bayes Classification as well as KDE models were performed [14] to predict students' admissions into universities or higher institutions based on a training dataset of students' examination scores. However, the approach did not investigate and enhancing the method to handle a wider selection of attributes with reduced complexity.

1.1 Novel Contributions

Motivated through above problems, such as, prediction accuracy, precision, prediction time, recall, in this work, Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier (KDODCNLC)is designed for predictive analytics.

- KDODCNLC has been developed to increase prediction accuracy through three distinct processes, all implemented within deep CNN.
- To minimize prediction time, the KDODCNLC model performs data preprocessing and feature selection. Rand similarity is applied in a max-pooling layer within the deep architecture to enhance feature extraction.
- Radial Kernel Discriminant Function is employed for analyzing the data samples, providing accurate classification results. The Rainfall Optimization Algorithm is used for fine-tuning, thereby minimizing errors in the classification process.
- Finally, the experimental analysis is performed to estimate result of proposed KDODCNLC model with different parameters.

1.2 Outline of manuscript

The organization of the paper is followed as. Section 2 reviews related works. In Section 3, explain the description of the proposed KDODCNLC method, including a detailed architecture diagram. The experimental analysis used for evaluation is provided as Section 4. In Section 5discusses the performance results of the KDODCNLC model and compares them with the outcome of existing methods. The conclusion of the paper is explained in Section 6.

2. Related Works

Data mining plays essential part at expansion of different areas, comprises education, security and so on. Free education, huge amount of applicants as well as elevated malfunction rates are challenging issues in education. To handle this issue, Data mining analyses were performed in [15] to examine the relationships among student result as well as admission aspects via Jaccard model. However, it failed to address on result of precision in predictions. ANN was designed in [16]. But, the time required for making predictions was not minimized. To handle this issue, ML and DL methods were developed [17] for predictive analytics at healthcare domain. However, it did not perform efficient analyses when dealing with massive amounts of streaming data. In addition, the university admission prediction was difficult based on the admission requirements.

University education is the key support of social and economic life. In [18], RF Regression method was designed for forecasting university admissions based on significant factors influencing the likelihood of admission. But, the handling of large datasets was not effective and did not incorporate a feature selection process. Yet another, ANN was developed [19] to forecast probability of student admittance based on different test attributes. However, the incorporation of advanced neural network architectures did not improve the prediction accuracy. In [20], ML methods were designed to predict admittance decisions based on decision-making process. But the time was higher.

With aim of improving predictions for higher education admissions, SVM stacking ensemble and PNN were developed in [21]. However, the approach failed to incorporate a feature selection model, which impacted its accuracy in predicting educational admissions. Yet another Ensemble machine learning algorithm was proposed in [22] to predict student performance. However, the specificity of the algorithms was not improved.

Logistic regression analysis was developed in [23] with lesser specificity to predict admissions more effectively. But, it failed to integrate additional aspect values, which failed to improve the prediction accuracy. A technique for forecasting intellectual performance of medical students in medical programs was developed in [24]. However, the precision was not concentrated. A Random Forest algorithm was designed in [25] for reliable and effective prediction of admission outcomes with higher precision. But, the deep learning techniques failed to address on the collection of larger datasets for application to enhance the university admission process.

Student selection in the health professions is a vital issue. Admission criteria as well as selection methods were developed in [26] for nursing students. However, these methods failed to improve effectiveness and large-scale feasibility. The decision support system developed in [27] for achieved a high level of accuracy in predicting student admissions using large data samples. In [28], the linear regression analysis was designed for predicting graduate admission chances. However, the designed method was not testing method on different database as well as explores the extra forecaster to improve prognostic accuracy.

To predict new student admissions, Naïve Bayes algorithm with a Gaussian distribution was developed in [29]. However, the model proved ineffective, resulting in significant prediction errors. A new model was designed in [30] for reducing error in predicting college and university admissions by incorporating clustering techniques. However, it failed to include sensitivity analysis in the prediction. A machine learning algorithm was designed in [31] to predict students' graduation in the Industrial Engineering course. However, it failed to improve model predictions when applied to a large volume of data. Also, the multiple operation conditions were not concentrated.

RUL forecast acts a crucial part in predictive maintenance. In [32], a method of three-phase FS method and DL model were designed for accurate prediction and RUL prediction. But, it failed to apply new methods to quick training process and improve the forecast outcomes. The Random Forest method, developed in [33], was designed to enhance feature selection for more effective classification data analysis. However, it failed to employ the improved predictive accuracy of method.

To enhance accuracy, an Ensemble Learning-basis of FS model was developed [34]. However, model did not optimize the weights of the ensemble learning components to minimize prediction error. Yet another feature selection and deep learning technique was developed in [35] for student result prediction with improved accuracy. However, various hybrid techniques were not implemented to enhance the feature selection process. Outline of existing studies summarized in Table 1.

Table 1 Overview of existing studies

Ref	Methodology	Objective	Advantages	Drawbacks
[1]	VR-ODRNN	To choose pertinent features	Minimum time	Huge dataset not considered
[2]	PSO-DNN	To achieve early prediction of student performance	Higher accuracy	Failed to consider admissions scheme
[3]	ANN	To forecast university performance	Enhance prediction performance	Multiclass classification was not carried out
[9]	Pattern discovery	To determine college programs for students	Achieve program prediction performance	Complexity was higher
[11]	K-means clustering	To classify Students' GPA	Higher precision	Significant features not selected
[18]	Random Forest Regression	To find university admissions	Lesser time	Failed to handle huge datasets
[22]	Ensemble machine learning	To obtain student performance	Recall enhanced	Lesser specificity
[27]	Decision support	To discover new student admission selection	Higher accuracy	Time was higher
[35]	Feature selection and deep learning	To predict student result	Higher precision	Failed to consider different hybrid methods

3. Proposal Methodology

The process of data mining is educating the precious data and patterns from big datasets using different techniques and algorithms. It combines elements of statistics, machine learning to analyze data. By employing various techniques and algorithms, the aim of data mining is to transform raw data into useful knowledge that used for decision-making and predictive analysis. As data continues to produce in volume and complexity, the significance of data mining in several industries will only improve.

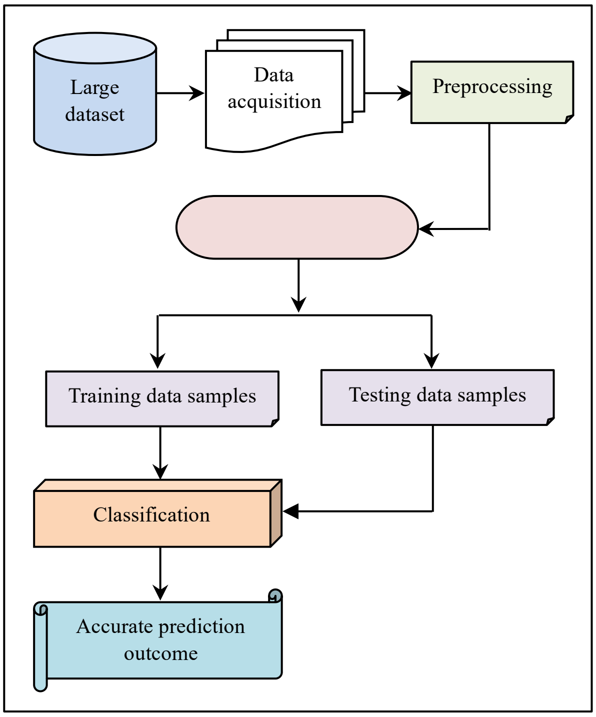

Figure 1: Architecture of KDODCNLC method

However, data mining also faces several limitations, including data quality issues, scalability challenges. Ensuring accurate, complete, and consistent data is crucial, while efficiently managing large datasets remains a significant hurdle. Addressing these limitations, a novel method called the KDODCNLC model is developed for achieving higher accuracy and minimum time complexity. KDODCNLC model comprises the different processes namely data acquisition, preprocessing, feature selection, classification, which are integrated into the Deep Convolutional Neural Learning Classifier.

3.1 Data Acquisition and Data Collection

Data acquisition refers to process of gathering data as of different sources of data for additional study, processing, storage. To gather information samples for predictive analysis, proposed technique uses Student Performance Prediction database considered from Kaggle (https://www.kaggle.com/datasets/souradippal/student-performance-prediction). This dataset aim is to predict student performance as pass or fail depend on different academic as well as demographic factors. Database presents challenges namely missing values, incorrect information, noise, making it ideal for practicing data cleaning, EDA, and feature engineering. Dataset includes 40,000 records of students and 7 features or attributes.

Table 2: Dataset Features and Description

S.No	**Features Name**	**Description**
1.	Student ID	A unique identifier for each student
2.	Study Hours per Week	Average number of hours student studies per week
3.	Attendance Rate	Percentage of classes a student attended

4.	Previous Grades	Average grade the student received in previous courses (scale of 0 to 100)
5.	Participation in Extracurricular Activities	Represents whether student participates in extracurricular activities (Yes/No)
6.	Parent Education Level	Highest level of education obtained with student's parents
7.	Passed	Target variable, representing whether the student passed the course (Yes/No)

The collected data samples and features is then stored in a suitable format for further processing. Let us consider the Student Performance Prediction dataset 'DS' comprise of data points 'DP' and features $f = \{f_1, f_2, \ldots, f_m\}$are organized in the form of matrix. Therefore, the input matrix with these dataset samples and features are formulated as follows,

$$M = \begin{bmatrix} f_1 & f_2 & \cdots & f_m \\ Dp_{11} & Dp_{12} & \cdots & Dp_{1n} \\ Dp_{21} & Dp_{22} & \cdots & Dp_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ Dp_{m1} & Dp_{m2} & \cdots & Dp_{mn} \end{bmatrix} \quad (1)$$

M indicates input matrix, $f_1, f_2, \ldots f_m$ represents number of features in column of matrix, row 'n' is denotes data samples or data that includes $Dp_1, Dp_2, \ldots Dp_n$. These input data are used for further processing in predictive analytics. Predictive analytics involves using DL methods to examine historical information and create forecast about upcoming events. The KDODCNLC model, through its structured approach, processes the input data using Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier provides accurate predictions. Below is a detailed description of each stage of the KDODCNLC model.

3.2 Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier

Convolutional Neural Learning Classifier is kind of DL model employed to processed data. It mainly valuable for reduced error rates and delivering accurate results, when handling huge volumes of data processing by the use of convolutional layers.

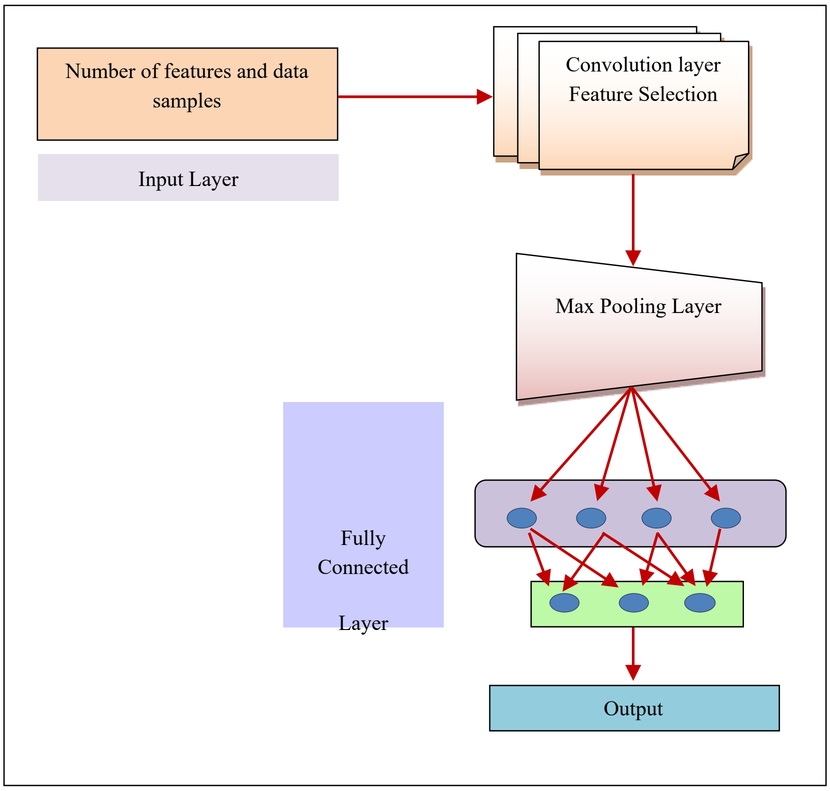

Figure 2: Schematic structure of DCNN classifier

The main aim of CNNs comprised their ability to learn hierarchical representations of features, which facilitates robust and accurate classification of data.

In Figure 2, describe the schematic structure of a DCNN classifier. This network comprises three main kinds of different layers namely input layer, one or more hidden layers, output layer. Single layers of input and output layers, whereas the hidden layers include multiple sub layers all of which are considered to process and utilize the input data. Each layer is made up of many artificial neurons to handle input samples as well as transmit the processed information to subsequent neurons in the following layers. Connections among neurons across different layers known as synapses. Final classification outcome is displayed at output layer.

Let assume a set of features '$f = \{f_1, f_2, \dots, f_m\}$' and, data points '$Dp = \{Dp_1, Dp_2, \dots, Dp_n\}$are given as input to initial layer of the DCNN classifier. In input layer, each neuron processes these data, with each neuron being associated with a set of weights. These weights are hyperparameters of NN to adjust and learn in the training process.

The neuron at input layer assigns weight sum of input data and the bias. This is given by the equation,

$$Q = \sum_{i=1}^{n} (Dp_i * w_i) + B \qquad (2)$$

Where, $Q$is an activity of the neuron at input layer, $w_i$represents weights allocated to input data 'Dp_i', 'B' is specified as a bias which saved value is '1'. The output of this weighted sum 'Q' is conceded through activation function.

$$\sigma_a(Q) = \left[1 + \frac{1}{\exp(-Q)}\right] \qquad (3)$$

Where, $\sigma_a$indicates a sigmoid activation function, Q denotes a neurons output. Neuron activity typically refers to the output of an artificial neuron (or node) in a neural network, which is usually within a specific range, often between 0 and 1. When the output is closer to 1 (or the upper bound of the activation function's), it indicates higher activity or activation of the neuron. When the output is closer to 0, it indicates lower activity. When a neuron is highly active (i.e., its output is close to 1), this indicates that the neuron is active. Then this output is transferred as input data to neurons at next layer of neural network.

- **Convolutional Layer**

The input data sample is transferred to hidden layer 1 (Convolutional layer), where preprocessing is carried out to remove unwanted data and handle the missing data. The weighted average method is determined for estimating missing values based on weighted average of available data values. The weighted average is expressed as,

$$Dp_m = \frac{\sum_{i=1}^{m} Dp_i * r_i}{\sum_{i=1}^{m} r_i} \qquad (4)$$

Where, Dp_m indicates a missing data, Dp_i denotes a known data points, r_i denotes a weight assigned to data 'Dp_i'.

Followed by, Dixon's test is employed for finding outlier information at provided database. It is employed to distinguish and categorize outliers as of database. First input information samples are arranged at ascending order. Dixon's test statistical test is expressed as follows,

$$DT = \frac{|Dp_i - Dp_{nn}|}{Mx_{Dp} - Mn_{Dp}} \qquad (5)$$

Where, DT indicates a Dixon's test, Dp_i indicates a data point in the particular cell, Dp_{nn} indicates nearest value of Dp_i, Mx_{Dp} indicates a maximum value in ascending order and the Mn_{Dp} indicates a minimum value in ascending order. The statistical test provides an output ranged from 0 to 1.

$$R = \begin{cases} DT > T & ; Outlier \\ DT < T & ; normal \end{cases} \qquad (6)$$

Where, R is an outcome, if calculated DT value is greater than threshold, the particular value is measured an outlier. If calculated DT value is lesser than or equal to threshold, the value is not measured an outlier and is considered normal data. As a result, the 4,005-outlier data are removed remaining 35,995 normal data points are considered for feature selection and the classification process.

- **Max Pooling Layers**

Through preprocessed data set, FS is carried out through using Max pooling layers. Max pooling layers are a type of pooling operation utilized in CNNs to minimize spatial dimensions of aspect maps whereas maintaining the most significant aspects. This operation helps in reducing the computational complexity. The Rand similarity is measured between the features.

$$S = 1 - \frac{|f_j \Delta f_N|}{n} \qquad (7)$$

Where, S indicates a Rand similarity measured between the features f_j and f_N, n is specified as a number of features. The Rand similarity provides output ranges from 0 to 1. The maxpooling operation finds the maximum similarity as,

$$Z = \max\{S\} \qquad (8)$$

Where, Z denotes maxpooling outcomes, $\max$ denotes a maximum function, S denotes a Rand similarity. As a result, the maxpooling operation selects the significant features with maximum similarity. Other aspects are extorted as of the database. Chosen output is sent into fully connected layers. Figure 3 shows the Rand similarity coefficient for 7 features before feature selection.

Student ID	Study Hours per Week	Attendance Rate	Previous Grades	Participation in Extracu	Parent Education Level	Passed
S00001	12.5	24	75.0	Yes	Master	Yes
S00002	9.3	27	60.6	No	High School	No
S00003	13.2	45	64.0	No	Associate	No
S00004	17.6	41	62.4	Yes	Bachelor	No
S00005	8.8	25	72.7	No	Master	No
S00006	8.8	55	69.3	Yes	High School	Yes
S00007	17.9	4	93.6	No	Doctorate	Yes
S00009	7.7	26	91.9	No	Bachelor	Yes
S00010	12.7	15	37.8	Yes	High School	No
S00011	7.7	59	72.3	No	Master	No
S00012	7.7	39	41.2	Yes	Master	No
S00013	11.2	22	49.6	Yes	Bachelor	No
S00014	0.4	31	50.4	No	No	Yes
S00015	1.4	9	49.2	No	Associate	Yes
S00016	7.2	34	55.9	Yes	High School	No
S00017	4.9	33	98.0	Yes	High School	No
S00018	11.6	28	51.8	No	Bachelor	Yes
S00019	5.5	21	40.8	No	Doctorate	No
S00020	2.9	33	72.4	No	Associate	Yes
S00021	17.3	8	68.6	No	Bachelor	Yes
S00022	8.9	19	98.2	No	Doctorate	No
S00023	10.3	15	64.4	No	High School	No
S00024	2.9	57	30.7	No	Bachelor	Yes
S00025	7.3	39	58.6	Yes	High School	Yes
S00026	10.6	40	58.3	No	Doctorate	Yes
S00027	4.2	31	46.5	No	High School	Yes
S00028	11.9	26	56.6	Yes	High School	No
S00029	7.0	44	62.8	Yes	Associate	No
S00032	19.3	33	74.1	Yes	Associate	Yes
S00033	9.9	57	89.5	No	Associate	No
S00034	4.7	38	69.2	No	Doctorate	Yes
S00035	14.1	53	81.6	No	High School	Yes
S00036	3.9	8	67.7	Yes	Master	No

Figure 3: Before feature selection outcome

```
Student ID      Rand Similarity : 0.74702125303967897

Study Hours per Week    Rand Similarity : 0.981412617610377

Attendance Rate Rand Similarity : 0.9967893730766949

Previous Grades Rand Similarity : 0.7248373972251683

Participation in Extracurricular Activities     Rand Similarity : 0.2639839122937674

Parent Education Level  Rand Similarity : 0.3433143279732907

Passed  Rand Similarity : 0.8640641724804313
```

Select relevant features data

Student ID	Study Hours per Week	Attendance Rate	Previous Grades	Passed
S00001	12.5	24	75.0	Yes
S00002	9.3	27	60.6	No
S00003	13.2	45	64.0	No
S00004	17.6	41	62.4	No
S00005	8.8	25	72.7	No
S00006	8.8	55	69.3	Yes
S00007	17.9	4	93.6	Yes
S00009	7.7	26	91.9	Yes
S00010	12.7	15	37.8	No
S00011	7.7	59	72.3	No
S00012	7.7	39	41.2	No
S00013	11.2	22	49.6	No
S00014	0.4	31	50.4	Yes
S00015	1.4	9	49.2	Yes
S00016	7.2	34	55.9	No
S00017	4.9	33	98.0	No
S00018	11.6	28	51.8	Yes
S00019	5.5	21	40.8	No
S00020	2.9	33	72.4	Yes
S00021	17.3	8	68.6	Yes
S00022	8.9	19	98.2	No
S00023	10.3	15	64.4	No
S00024	2.9	57	30.7	Yes
S00025	7.3	39	58.6	Yes
S00026	10.6	40	58.3	Yes
S00027	4.2	31	46.5	Yes
S00028	11.9	26	56.6	No
S00029	7.0	44	62.8	No
S00032	19.3	33	74.1	Yes
S00033	9.9	57	89.5	No
S00034	4.7	38	69.2	Yes
S00035	14.1	53	81.6	Yes
S00036	3.9	8	67.7	No

Figure 4: After feature selection outcome

In the maxpooling layer, before the feature selection outcome, 7 features were considered, such as such as, Student ID, Study Hours per Week, and so on and Passed to estimate Rand similarity.

As shown in Figure 4, only the relevant features are selected after feature selection outcome by using Rand similarity. The relevant features such as Student ID, Study Hours per Week, Attendance Rate, Previous Grades and Passed are chosen.

- **Fully Connected Layers**

Itis also called dense layers, where every neuron is associated to each neuron at preceding layer. This structural design is typically used at final layer of a CNN, mainly feed-forward neural networks and deep learning models. Fully Connected Layers are utilized for predictions based on selected features from the maxpooling layer.

KDA is an extension of Linear Discriminant Analysis (LDA) to incorporate kernel methods to handle selected features. This allows for modeling more complex relationships among testing as well as training information samples. Radial kernel is measured based on testing as well as training information samples at input space.

$$K\ (Dp_t, Dp_t) = \exp\left[-0.5 * \left(\frac{|Dp_t - Dp_{tr}|^2}{\sigma^2}\right)\right] \qquad (9)$$

Where, K denotes an outcome of kernel output, Dp_t denotes a testing information samples, Dp_{tr} represents training information samples, 'σ'specifies standard deviation. Depend on kernel output, input data samples are categorized. These classification results are used for accurate predictions with minimal time consumption.

To reduce error, Fine-tuning process employed in the deep CNN involves adjusting hyperparameters, namely weights. This process is commonly achieved through the use of the

Rain-fall optimization algorithm. The Rain-fall optimization is a nature-inspired metaheuristic algorithm that mimics the performance of raindrops falling from a hill and always looking for the lowest points or valleys leading to the sea. In this algorithm, the raindrops are related to the number of weights between the layers.

$$w_i = w_1, w_2, w_3, \ldots, w_n \qquad (10)$$

Where, w_i indicates weight among the layers. After that, fitness is estimated for each weight.

$$F = \arg\min(ER) \qquad (11)$$

Where, ER denotes a error rate, $\arg\min$ denotes a argument of minimum error, F indicates a fitness,

$$ER = (A_O - P_O)^2 \qquad (12)$$

Where, A_O denotes an actual outcome, P_O denotes a predicted outcome. As a result, the raindrop with better fitness is selected as the active and others it is called as inactive raindrops. In other words, the weight which satisfies the fitness (i.e. minimal error) is selected. The position of selected active raindrops is updated according to the following procedure:

$$P_{t+1} = (1 - \vartheta) * r(-1,1) * d_f * P_t + \vartheta * r(-1,1) * X_s * P_{best} \qquad (13)$$

Where, P_{t+1} denotes an updated position, 'ϑ 'denotes a weight of raindrop, $r(-1,1)$ random numbers, d_f indicates a contraction factor of the current position of raindrop P_t, X_s denotes a search factor of the optimal raindrop 'P_{best}'.

If the fitness of the updated position is better than that of the previous position, it replaces the current best solution. This process continues until the maximum number of iterations is reached. Finally, the optimal solution (i.e., optimal weights) is selected to minimize classification error. After the fine-tuning process, the final prediction output with minimal error is obtained at the output layer. The research focused on identifying an accurate prediction of students and providing timely interventions to use predictive analytics and improve their academic outcomes.

Algorithm 1 for Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier describes the processes involved in prediction using a Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier. The number of data samples and features is provided as input to the input layer. For each input, weights and bias functions are assigned by the neuron. Then, the input is transferred to the neurons in the hidden layer based on the activation function outcome. In the convolutional layer, data preprocessing is performed through missing data handling and outlier data removal. Following this, the significant feature selection process is carried out in the max-pooling layer. Finally, the classification process is done at the fully connected layer by applying the kernel function. After classification, the error rate is calculated based on the actual and predicted output results.

Algorithm 1: Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier
Input: Datasets 'DS', data points '$Dp = \{Dp_1, Dp_2, \ldots, Dp_n\}$', Features '$f = \{f_1, f_2, \ldots, f_m\}$' **Output:** Increase the prediction accuracy
Begin

1. **Number of** Features '$\{f_1, f_2, \ldots, f_m\}$'with data samples $Dp_1, Dp_2, \ldots, Dp_n$ taken **at the input layer**
2. **For each** data Dp **- [hidden layer]**
3. Formulate the neuron activity using (2) (3)
4. **End for**
5. Perform data preprocessing using (4) (5) - **[convolutional layer]**
6. Select the significant features using (7) (8) - **[maxpooling layer]**
7. **For each** training data **w**ith **t**esting data - **[fully connected layer]**
8. Measure relationship using kernel function (9)
9. Classify the data samples
10. **End for**
11. **For each classified result**
12. Measure the error rate 'E'
13. Update the weight
14. Initialize the populations of raindrops (i.e. weights)
15. **For each** raindrop
16. **While** (t< maximum number of iterations) do
17. Calculate the fitness using (11)
18. $\boldsymbol{if}\,(\arg\min ER)$ **then**
19. Select active raindrop
20. **Else**
21. Select inactive raindrop'
22. **End if**
23. Update the positions of each raindrop using (13)
24. Replace current best solution
25. Increment **t= t+1**
26. Go to step 16
27. **End While**
28. **End for**
29. **End for**
30. Obtain the best solution
31. Display the final classification results with minimum error **at the output layer**

End

To minimize this error, the Rainfall Optimization algorithm is applied to find the optimal weight value. Subsequently, the active or inactive raindrops are selected based on the fitness function. This entire process iterates until the algorithm reaches its maximum iterations. This iterative approach enables the algorithm to identify the optimal weight value to minimize the error rate and improve the accuracy of prediction. Finally, the optimal weight value is determined to minimize the error rate of the entire classifier.

4. Experimental analysis

The proposed Kernel Discriminant Optimized Deep Convolutional Neural Learning Classifier (KDODCNLC) is evaluated employing Python high-level general-purpose programming language, working on a computer Intel(R) Core (TM) i7-6700HQ CPU@2.60GHz with a RAM of 32 GB running Windows 10. The results are compared with the previous three existing methods, namely PSO-DNN [1], VR-ODRNN [2], and ANN [3].Moreover, the results are analyzed and validated on the basis of six metrics such as precision, recall, prediction accuracy, prediction time, F1-score, and specificity using the Student Performance Prediction dataset obtained from https://www.kaggle.com/datasets/souradippal/student-performance-prediction. In addition, for evaluation, the maximum number of data samples is taken as 50000. To ensure fair comparisons, sample data from the same Student Performance Prediction dataset is applied to the four methods, KDODCNLC, PSO-DNN [1], VR-ODRNN [2], and ANN [3], and validated, analyzed for an average of 10 simulation runs.

5. Discussion

The results of the proposed KDODCNLC and existing PSO-DNN [1], VR-ODRNN [2], and ANN [3] are discussed using several parameters with the aid of tables and graphical representation.

5.1 Performance analysis of Precision

In this section, the performance analysis of precision is validated. It measures the proportion of true positive predictions among all positive predictions made by the model. Therefore, precision is formulated as follows,

$$Pre = \left(\frac{TP}{TP+FP}\right) * 100 \qquad (14)$$

From the above equations (15), precision 'Pre' is estimated employing true positive rate 'TP' and false positive rate 'FP'. Table 3 given below lists the precision analyses using KDODCNLC and PSO-DNN [1], VR-ODRNN [2], and ANN [3].

Table 3: Tabulation of Precision using KDODCNLC and PSO-DNN [1], VR-ODRNN [2], and ANN [3]

Number of data samples	Precision (%)			
	VR-ODRNN	PSO-DNN	ANN	KDODCNLC
5000	96.75	94.01	92.19	98.06
10000	94.45	92.35	90.22	97.15
15000	93.66	91.33	89.19	96.32
20000	93.88	91.68	89.42	96.58
25000	94.66	92.05	89.75	97.11
30000	95.08	92.33	90.37	97.04

35000	93.56	91.45	89.25	95.66
40000	94.05	92.2	89.45	96.46
45000	94.65	92.35	89.69	96.89
50000	95.11	92.75	90.48	97.25

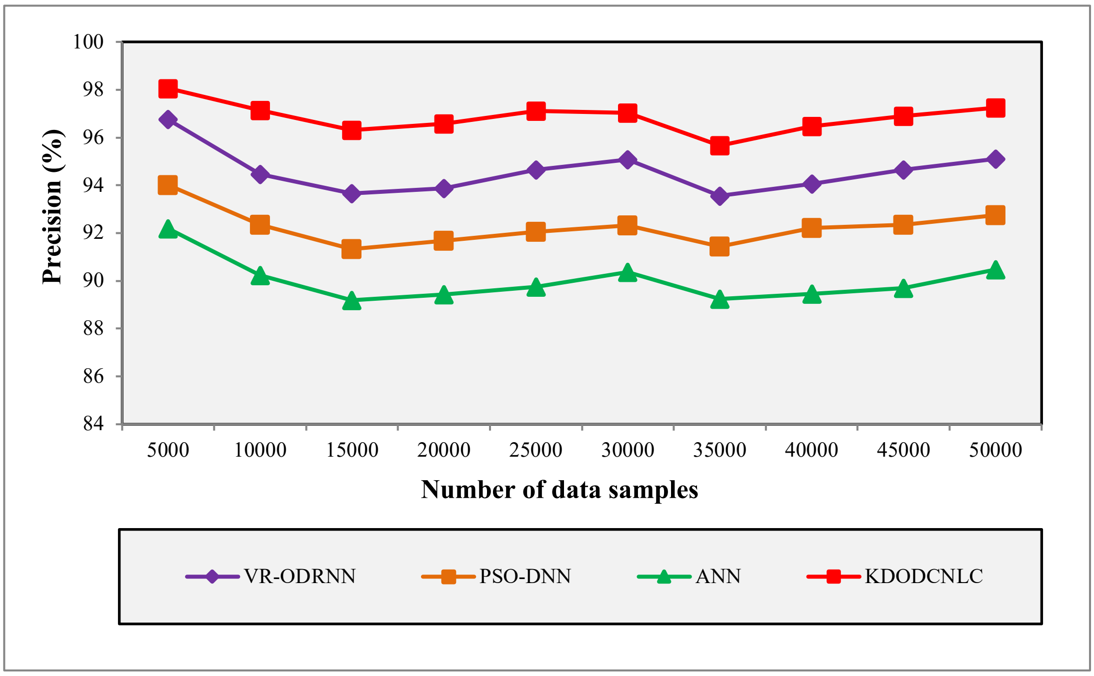

Figure 5: Performance outcomes of Precision

Figure 5 shows the graphical representation of precision. The precision is computed by applying four methods, namely KDODCNLC with three existing methods, PSO-DNN [1], VR-ODRNN [2], and ANN [3]. Let us consider the first iteration involving 5000 samples from the dataset, the precision using the KDODCNLC was found to be 98.06%. Subsequently, 96.75%, 94.01% and 92.19% of precision were observed by applying [1] [2], and [3], respectively. As illustrated in the above figure, the precision using KDODCNLC was found to be comparatively better than [1], [2], and [3]. The higher precision is achieved by using Deep Convolutional Neural Learning classifier. Relevant features (Student ID, Attendance Rate, and Passed) are extracted by using max-pooling layer. Also, the data samples are accurately categorized by applying the radial Kernel Discriminant function with lesser incorrect classifications and higher classification accuracy. The overall comparison result indicates that the proposed **KDODCNLC** increased precision by 2% compared to [1] and 5% compared to [2] and 8% compared to [3], respectively.

5.2 Study of Sensitivity or Recall

Sensitivity or Recall is used to measure the ability of a model to correctly identify all positive cases. The recall is mathematically represented as given below.

$$Rec = \left(\frac{TP}{TP+FN}\right) * 100 \quad (15)$$

From the above equation (15), recall rate 'Rec' is measured based on the true positive rate 'TP' and false negative rate 'FP'. Table 4 given below lists the recall analyses using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3].

Table 4: Tabulation of Recall using KDODCNLC and PSO-DNN [1], VR-ODRNN [2], and ANN [3]

Number of data samples	Recall (%)			
	VR-ODRNN	PSO-DNN	ANN	KDODCNLC
5000	98.82	98.43	97.92	99.38
10000	96.32	95.45	93.42	97.56
15000	96.13	95.12	93.34	97.12
20000	96.08	95.02	93.23	97.06
25000	96.45	94.65	92.65	97.39
30000	96.17	94.54	92.33	97.65
35000	96.01	94.45	92.06	97.48
40000	95.35	93.05	91.65	96.62
45000	95.65	93	91.47	96.86
50000	95.82	93.36	92.09	97.18

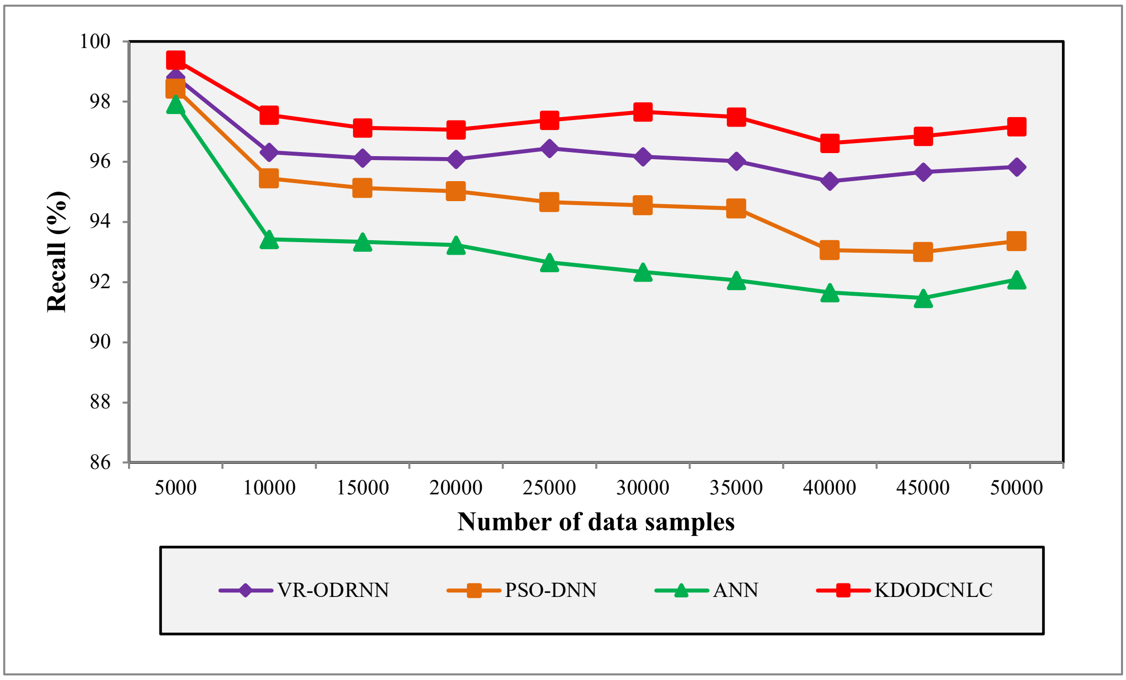

Figure6: Performance outcomes of Recall

Figure 6 demonstrates the result analysis recall based on the number of data samples using four methods. The number of data samples is taken in the horizontal direction, where the results of recall for four methods are obtained in the vertical direction. Also, simulations performed with 5000 numbers of data samples observed the true positive rate of 4820, 4620, 4400, and 4250 and false positive rate of 30, 55, 70, and 90 by applying KDODCNLC, PSO-DNN [1], VR-ODRNN [2], and ANN [3], respectively. With this, the overall recall rate was observed to be 99.38%, 98.82%, 98.43% and 97.92% respectively. As exposed in the above graphical representation, the proposed KDODCNLC takes a higher recall as compared to other classification methods. The KDODCNLC model reduces the error between the actual and predicted outputs by applying Rainfall Optimization to update the classifier's hyperparameters, thus minimizing false negative rates in the predictions. Therefore, the recall using KDODCNLC by 2% compared to [1] and 3% compared to [2] and 5% compared to [3], respectively.

5.3 Impact of Prediction accuracy

Prediction accuracy plays a major role in Student performance prediction analysis. The prediction accuracy performance metrics are mathematically represented as given below.

$$PA = \left(\sum_{i=1}^{N} \frac{AP}{Dp_i}\right) * 100 \quad (16)$$

From the above equation (16), accuracy 'A' is measured based on the number of data samples involved in the simulation process 'Dp_i' and the data samples properly predicted 'AP'. Both accuracy and precision performance metrics are measured in terms of percentage (%). Table 5 given below lists the prediction accuracy analyses using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3].

Table 5: Tabulation of Prediction Accuracy using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3]

Number of data samples	Prediction Accuracy (%)			
	VR-ODRNN	PSO-DNN	ANN	KDODCNLC
5000	95.86	93.74	89.36	97
10000	95.05	93.05	89.55	96.05
15000	94.45	92.05	88.15	95.58
20000	94.68	92.42	88.62	95.86
25000	94.25	92.22	88.33	96.05
30000	93.74	91.05	87.65	96.33
35000	94.03	92.47	88.15	95.74
40000	93.78	91.45	86.35	96.65
45000	94.11	92.36	87.16	96.08
50000	93.08	91.07	86.48	95.24

Figure 7 given below explains the graphical representations of prediction accuracy for 5000-50000 distinct numbers of data samples. This is evident from the simulation with 5000 sample, 48504793, 4686 and 4468 images were accurately predicted using KDODCNLC, [1], [2], and [3]. The accuracy using the KDODCNLC was found to be 97%. Subsequently, 95.86%, 93.74% and 89.36% of accuracy were observed by applying [1] [2] and [3], respectively. As illustrated in the above figure, the prediction accuracy using the proposed KDODCNLC was found to be comparatively better than [1] [2] and [3]. The reason for higher accuracy is achieved by the Kernel Discriminant Optimized Deep Convolutional Neural Learning. The feature selection employed to choose five relevant features, such as Student ID, Study Hours per Week, Attendance Rate, Previous Grades and Passed **to enhance the accuracy. Also, the classifier uses the Kernel Discriminant function to analyze the testing and training data samples, which helps to improve the true positive rate. Additionally, the Rainfall Optimization algorithm is used for hyperparameter optimization, minimizing classification error, and increasing prediction accuracy. The average of the comparison of ten results reveals that the prediction accuracy performance of KDODCNLC was increased by 2% compared to [1] and 4% compared to [2] and 9% compared to [3].**

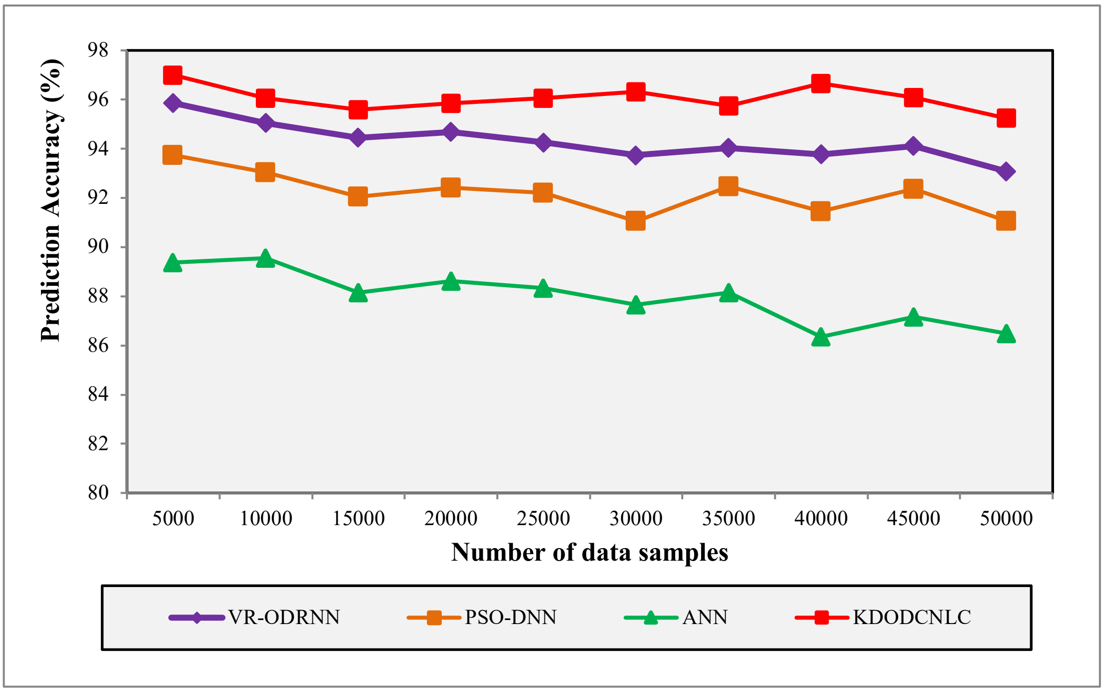

Figure7: Performance outcomes of Prediction Accuracy

5.4 Result of Prediction time

A proportionate amount of time is said to be used for accurate prediction, and this is referred to as **Prediction time**. It is measured as given below.

$$PT = \sum_{i=1}^{n} Dp_i * Time(AP) \qquad \textbf{(17)}$$

From the above equation (17), the prediction time 'PT' is evaluated by taking into consideration the data samples 'Dp_i' involved in simulation process and the time consumed in precise prediction '$Time\ (AP)$'. It is measured in milliseconds (ms). Table 6 given below lists the prediction time analyses using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3].

Table 6: Tabulation of Prediction time using KDODCNLC and PSO-DNN [1], VR-ODRNN [2], and ANN [3]

Number of data samples	Prediction time (s)			
	VR-ODRNN	PSO-DNN	ANN	KDODCNLC
5000	28.36	33.22	40.44	24.28
10000	33.58	38.52	44.18	28.64
15000	37.82	42.63	49.39	30.42
20000	40.54	47.87	56.24	34.66
25000	42.38	52.53	59.55	38.82
30000	48.69	55.61	62.22	40.25
35000	55.63	60.92	69.47	45.68
40000	60.82	65.66	76.48	50.63
45000	77.92	82.78	90.78	58.39
50000	80.31	92.6	105.4	60.54

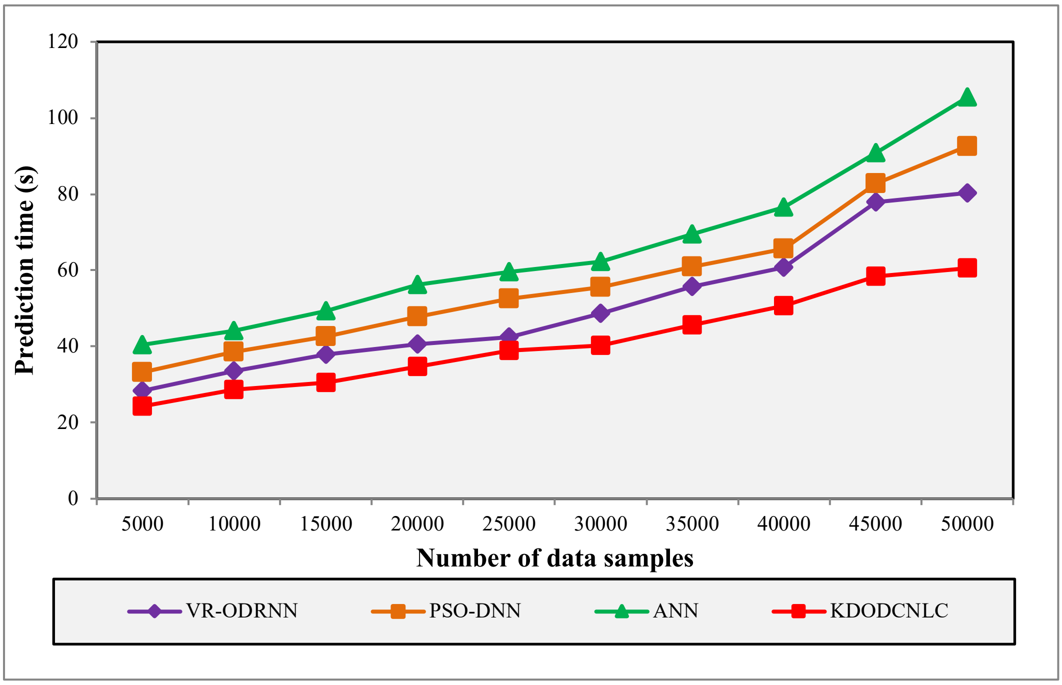

Figure 8: Performance outcomes of Prediction Time

Figure 8 exemplifies performance analysis on prediction time values for ten different numbers of data samples. The graphical analysis prediction time is performed on four methods using the Proposed KDODCNLC and existing PSO-DNN [1], VR-ODRNN [2], and ANN [3]. However, simulations performed with 5000 data samples, observed prediction time of 24.28s using the proposed KDODCNLC whereas 28.36s, 33.22s, and 40.44s using [1] and [2]. Among diverse methods, the KDODCNLC minimized the prediction time compared to other methods.

The reduced prediction time of the KDODCNLC model helps in its efficient data preprocessing and feature selection processes, which minimize the number of features and the data processed, thereby increasing prediction accuracy and reducing the time consumption. Accordingly, the prediction time examines that the proposed KDODCNLC reduces the prediction time by 17%, 27% and 37% when compared to [1], [2], and [3], respectively.

5.5 Performance of F1-Score

It is a performance metric that combines both precision and recall into a single value. It is formulated as follows:

$$F1 - Score = 2 * \left(\frac{Pre * Rec}{Pre + Rec}\right) \quad (18)$$

From the above equation (18), Rec denotes a recall, Pre denotes the precision. Table 7 given below lists the F1-Score analyses using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3].

Table 7: Tabulation of F1-Score using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3]

Number of data samples	F1-Score (%)			
	VR-ODRNN	PSO-DNN	ANN	KDODCNLC
5000	97.77	96.16	94.96	98.71
10000	95.37	93.87	91.79	97.35
15000	94.87	93.18	91.21	96.71
20000	94.96	93.32	91.28	96.81
25000	95.54	93.33	91.17	97.24
30000	95.62	93.42	91.33	97.34
35000	94.76	92.92	90.63	96.56
40000	94.69	92.62	90.53	96.53
45000	95.14	92.67	90.57	96.87
50000	95.46	93.05	91.27	97.21

Figure 9 displays the performance outcomes of the F1-score based on the number of data samples. The number of data samples ranges from 5000 to 50000. According to the observed results, the proposed KDODCNLC demonstrated improved performance of the F1-Score in accurate prediction compared to existing models. This is because of the KDODCNLC enhances the performance of both precision and recall in the prediction. F1-Score using the KDODCNLC is improved by 2%, 4% and 6% compared to [1] [2], and [3], respectively.

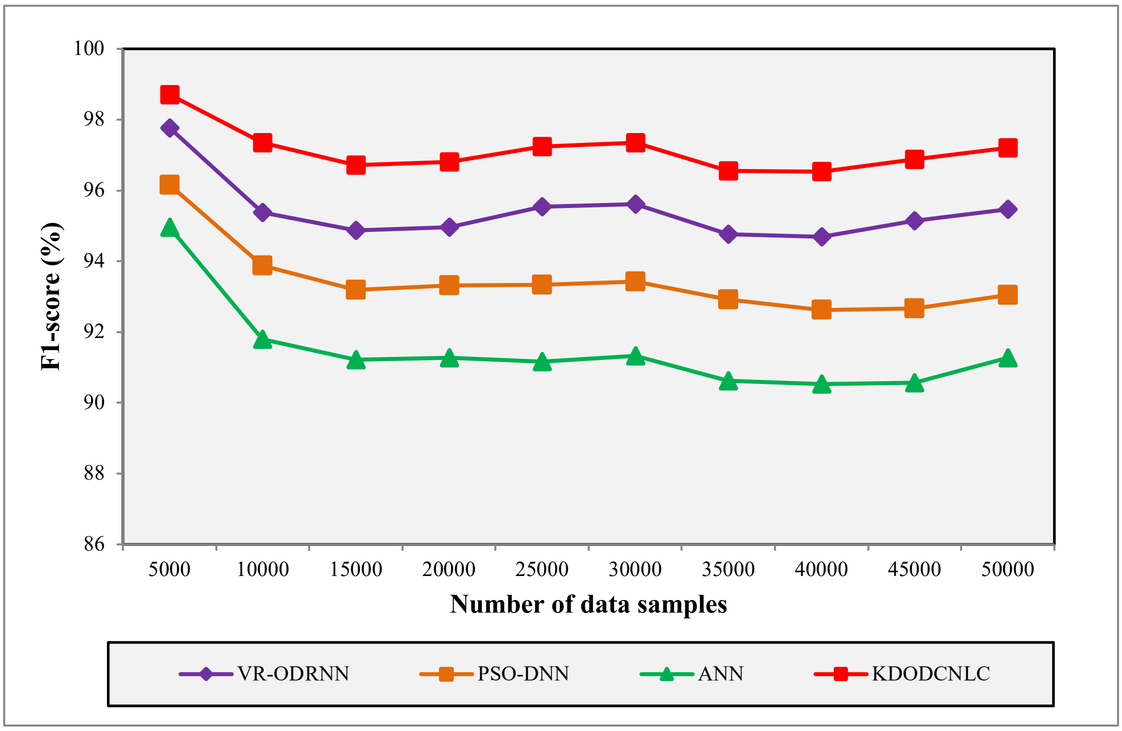

Figure 9: Performance outcomes of F1-Score

5.6 Outcome of Specificity

It is described as the amount of improperly predicted data samples divided by the total quantity of data samples. It is calculated as follows:

$$Specificity = \left(\sum_{i=1}^{N} \frac{IP}{Dp_i}\right) * 100 \quad (19)$$

From the above equation (19), the number of data samples involved in the simulation process 'Dp_i' and the data samples improperly predicted 'IP'. Table 8 given below lists the Specificity analyses using KDODCNLC and PSO-DNN [1], VR-ODRNN [2] and ANN [3].

Table 8: Tabulation of Specificity using KDODCNLC and PSO-DNN [1], VR-ODRNN [2], and ANN [3]

Number of data samples	Specificity (%)			
	VR-ODRNN	PSO-DNN	ANN	KDODCNLC
5000	64.36	60.22	52.82	68
10000	64.09	60.01	52.05	68.32
15000	61.35	58.08	51.02	68.49
20000	61.78	58.32	51.22	68.52
25000	63.45	59.44	51.6	69.55
30000	62.86	60.02	55.04	68.29
35000	62.06	60.28	55.49	67.38
40000	62.67	58.55	52.43	67.65
45000	63.22	59.16	52.32	66.32
50000	62.05	59.08	52.09	65.11

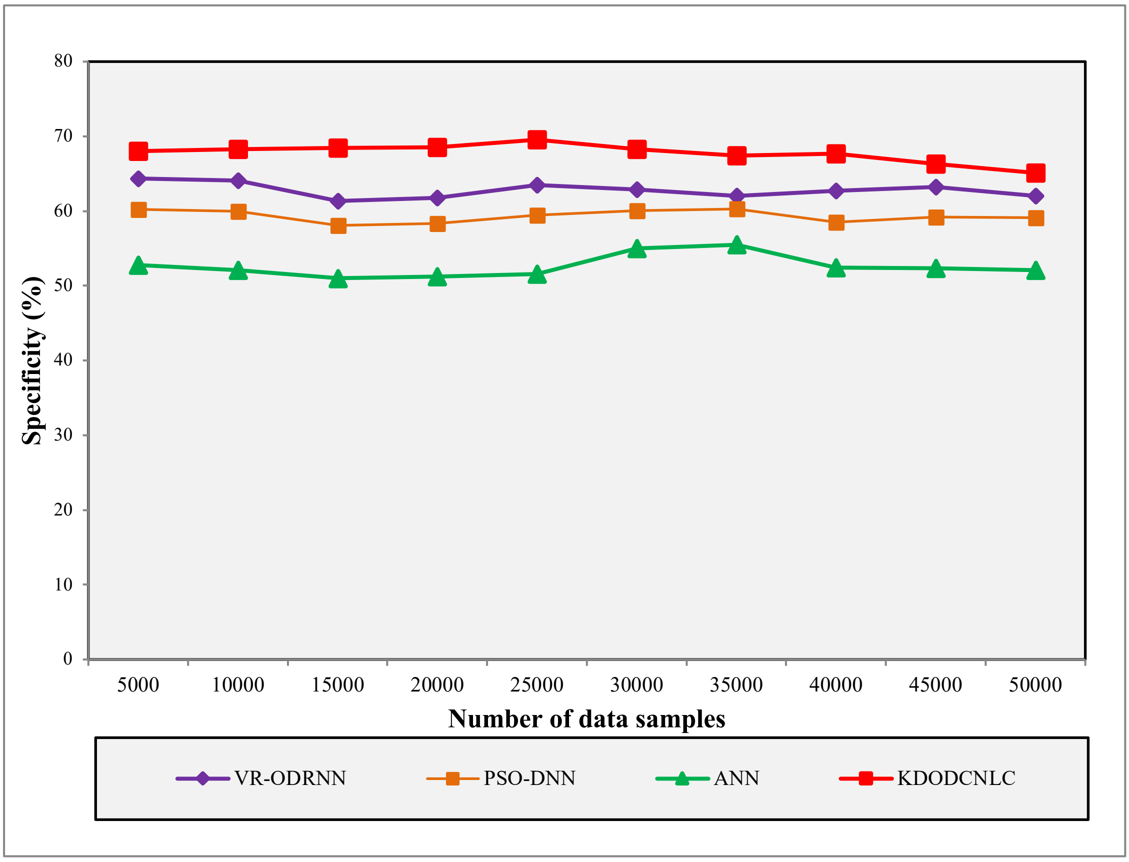

Figure 10: Performance outcomes of Specificity

Figure 10 shows the performance analysis on specificity. The graphical analysis on specificity is carried out using four methods through the proposed KDODCNLC and existing IDLQET-BTEDC model, existing KDODCNLC and PSO-DNN [1], VR-ODRNN [2], and ANN [3]. The higher specificity is achieved by means of feature selection and accurate classification. As a result, the specificity of the KDODCNLC model improved by 8% compared to [1] and by 14% compared to [2] and 29% compared to [3].

6. Conclusion

In this study, an efficient deep learning model named KDODCNLC is proposed for accurate prediction with the objective of improving precision, accuracy, and reducing time. First, data samples and features are obtained from the Student Performance Prediction dataset. Then, preprocessing and feature selection are applied in KDODCNLC by handling missing data and selecting applicable features from the dataset. Followed by, classification is performed with a kernelized discriminant function for providing the prediction outcomes in an accurate manner. Also, the Rain-fall algorithm is employed for hyperparameter optimization to reduce the error rate in the prediction process. To validate the proposed and existing methods, the experiment is conducted using different evaluation metrics. The overall performance results illustrate that the presented KGHF-CCDTL method achieves higher accuracy up to 5% with minimum time by 27% than the conventional methods.

References

[1] Iyad Katib, Mahmoud Ragab, "Harnessing variable reduction approach with deep recurrent neural network for student's academic performance analysis", Alexandria, Volume 118, 2025, Pages 393-405. https://doi.org/10.1016/j.aej.2025.01.082

[2] Ahmet Kala, Orhan Torkul, Tugba Tunacan Yildiz, Ihsan Hakan Selvi, "Early Prediction of Student Performance in Face-to-Face Education Environments: A Hybrid Deep Learning Approach With XAI Techniques", IEEE Access, Volume 12, 2024, Pages 191635 – 191649. DOI: 10.1109/ACCESS.2024.3516816

[3] Hanan Abdullah Mengash, "Using Data Mining Techniques to Predict Student Performance to Support Decision Making in University Admission Systems", IEEE Access, Volume 8, 2020, Pages 55462-55470. DOI: 10.1109/ACCESS.2020.2981905

[4] Vimarsha K, S.P. Shiva Prakash, Kirill Krinkin, YuliaA. Shihckina, "Student Performance Prediction: A Co-Evolutionary Hybrid Intelligence model", Procedia Computer Science,Volume 235, 2024, Pages 436-446. https://doi.org/10.1016/j.procs.2024.04.043

[5] Kuburat Oyeranti Adefemi and Murimo Bethel Mutanga, "A Robust Hybrid CNN–LSTM Model for Predicting Student Academic Performance", Digital, MDPI, Volume 5, Issue 2, 2025, Pages 1-15. https://doi.org/10.3390/digital5020016

[6] Kuburat Oyeranti Adefemi, Murimo Bethel Mutanga and Vikash Jugoo, "Hybrid Deep Learning Models for Predicting Student Academic Performance", Mathematical and Computational Applications, MDPI, Volume30, Issue 3, Pages 1-18. https://doi.org/10.3390/mca30030059

[7] Emi Kalita, Abdullah Mana Alfarwan, Houssam El Aouifi, Ashima Kukkar, Sadiq Hussain, Tazid Ali and Silvia Gaftandzhieva, "Predicting student academic performance using Bi-LSTM: a deep learning framework with SHAP-based interpretability and statistical validation", Frontiers in Education, Volume 10, 2025, Pages 1-18. https://doi.org/10.3389/feduc.2025.1581247

[8] Essa Alhazmi, Abdullah Sheneamer, “Early Predicting of Students Performance in Higher Education”, IEEE Access, Volume 11, 2023, Pages 27579 – 27589.DOI: 10.1109/ACCESS.2023.3250702
[9] Julius Cesar O. Mamari, Melvin A. Ballera, “Multiple educational data mining approaches to discover patterns in university admissions for program prediction”, International Journal of Informatics and Communication Technology, Volume 11, Issue 1, 2022, Pages 45-56. http://doi.org/10.11591/ijict.v11i1.pp45-56
[10] Ayman Qahmash, Naim Ahmad and Abdulmohsen Algarni, “Investigating Students’ Pre-University Admission Requirements and Their Correlation with Academic Performance for Medical Students: An Educational Data Mining Approach”, Brain Sciences, Volume 13, Issue 3, 2023, Pages 1-12. https://doi.org/10.3390/brainsci13030456
[11] Raden Gunawan Santosa, Yuan Lukito, Antonius Rachmat Chrismanto, “Classification and Prediction of Students’ GPA Using K-Means Clustering Algorithm to Assist Student Admission Process”, Journal of Information Systems Engineering and Business Intelligence, Volume 7, Issue 1,2021, Pages 1-10. https://doi.org/10.20473/jisebi.7.1.1-10
[12] Julius Cesar O. Mamaril, “Predictive and Forecasting Models for University Admissions and Graduation Performance: Leveraging Data Mining Algorithms”, Southeast Asian Journal of Science and Technology, Volume 6, Issue 2, 2021, Pages 1-6. https://www.sajst.org/online/index.php/sajst/article/view/275
[13] Adrita Iman, Xiaoguang Tian, “A Comparison of Classification Models in Predicting Graduate Admission Decision”, Journal of Higher Education Theory and Practice, Volume 21 Issue 7, 2021, Pages 219-230. https://doi.org/10.33423/jhetp.v21i7.4498
[14] Nasim Matar, Wasef Matar, Tirad Al Malahmeh, “A Predictive Model for Students Admission Uncertainty Using Naïve Bayes Classifier and Kernel Density Estimation (KDE)”, International Journal of Emerging Technologies in Learning (iJET), Volume 17, Issue 8, Pages 75-96. https://www.learntechlib.org/p/222830/
[15] Basem Assiri, Mohammed Bashraheel and Ala Alsuri, “Enhanced Student Admission Procedures at Universities Using Data Mining and Machine Learning Techniques”, Applied Sciences, Volume 14, Issue 3, Pages 1-19.
https://doi.org/10.3390/app14031109
[16] Anietie Ekong, Abasiama Silas, Saviour Inyang, “A Machine Learning Approach for Prediction of Students’ Admissibility for Post-Secondary Education using Artificial Neural Network”, International Journal of Computer Applications, Volume 184, Issue 27, 2022, Pages 44-49. DOI:10.5120/ijca2022922340
[17] Mohammed Badawy, Nagy Ramadan and Hesham Ahmed Hefny, “Healthcare predictive analytics using machine learning and deep learning techniques: a survey”, Journal of Electrical Systems and Information Technology, Springer, Volume 10, 2023, Pages 1-45. https://doi.org/10.1186/s43067-023-00108-y
[18] Inssaf El Guabassi, Zakaria Bousalem, Rim Marah, Aimad Qazdar, “A Recommender System for Predicting Students' Admission to a Graduate Program using Machine Learning Algorithms”, International Journal of Online and Biomedical Engineering, Volume 17, Issue 02, Pages 135-147. https://doi.org/10.3991/ijoe.v17i02.20049
[19] Meenakshi Bhrugubanda, Varsha Udutha, Sri Charan Yella, “Post Graduate Admission Prediction Using ANN”, International Journal for Research in Applied Science & Engineering

Technology, Volume 11, Issue 6, 2023, Pages 219-222. https://doi.org/10.22214/ijraset.2023.53463
[20] Yijun Zhao, Xiaoyu Chen, Haoran Xue, Gary M. Weiss, "A machine learning approach to graduate admissions and the role of letters of recommendation", PLoS ONE, Volume 18, Issue 10, 2023, Pages 1-17.
https://doi.org/10.1371/journal.pone.0291107
[21] Khrystyna Zub, Pavlo Zhezhnych and Christine Strauss, "Two-Stage PNN–SVM Ensemble for Higher Education Admission Prediction", Big Data Cognitive Computing, 2023, Volume 7, Issue 2, Pages 1-13. https://doi.org/10.3390/bdcc7020083
[22] Mahmoud Ragab, Ahmed M. K. Abdel Aal, Ali O. Jifri, and Nahla F. Omran, "Enhancement of Predicting Students Performance Model Using Ensemble Approaches and Educational Data Mining Techniques", Wireless Communications and Mobile Computing, Hindawi, Volume 2021, December 2021, Pages 1-9. https://doi.org/10.1155/2023/9813635
[23] Jung-Pin Wu, Ming-Shr Lin and Chi-Lun Tsai, "A Predictive Model That Aligns Admission Offers with Student Enrollment Probability", Education Sciences, Volume 13, Issue 5, 2023, Pages 1-13. https://doi.org/10.3390/educsci13050440
[24] Amer Almarabheh, Mohamed Hany Shehata, Abdulrahman Ismaeel, Hani Atwa and Ahmed Jaradat, "Predictive validity of admission criteria in predicting academic performance of medical students: A retrospective cohort study", Frontiers in Medicine, Volume 9, 2022, Pages 1-10.
https://doi.org/10.3389/fmed.2022.971926
[25] Aga Maulana, Teuku Rizky Noviandy, Novi Reandy Sasmita, Maria Paristiowati, Rivansyah Suhendra, Erkata Yandri, Justinus Satrioand Rinaldi Idroes, "Optimizing University Admissions: A Machine Learning Perspective", Journal of Educational Management and Learning, Volume 1, Issue 1, 2023, Pages 1-7.
https://doi.org/10.60084/jeml.v1i1.46
[26] Vahid Zamanzadeh, Akram Ghahramanian, Leila Valizadeh, Farzaneh Bagheriyeh & Marita Lynagh, "A scoping review of admission criteria and selection methods in nursing education", BMC Nursing, Elsevier, Volume 19, 2020, Pages 1-17.
https://doi.org/10.1186/s12912-020-00510-1
[27] Yunita Sari Siregar, Ahmad Zakir, Nenna Irsa Syahputri, Herlina Harahap, Divi Handoko, "Analysis of Decision Support Systems Edas Method in New Student Admission Selection", Journal of Computer Networks, Architecture and High-Performance Computing, Volume 5, Issue 1, 2023, Pages 251-262.
https://doi.org/10.47709/cnahpc.v5i1.2057
[28] Ashiqul Haque Ahmed, Sabbir Ahmad, Md Abu Sayed, Malay sarkar, Eftekhar Hossain Ayon, Md Tuhin Mia, Tahera Koli and Rumana Shahid, "Predicting the Possibility of Student Admission into Graduate Admission by Regression Model: A Statistical Analysis", Journal of Mathematics and Statistics Studies, Volume 4, Issue 4, 2023, 97-105, https://doi.org/10.32996/jmss.2023.4.4.10
[29] Aulia Salsabila, Marnis Nasution, Irmayanti, "Implementation of the Naïve Bayes Algorithm to Predict New Student Admissions", Building of Informatics, Technology and Science, Volume 6, Issue 1, 2024, Page 421-429.
https://doi.org/10.47065/bits.v6i1.5363

[30] Xiao Chen, Yi Peng, Yachun Gao, Shimin Cai, "A competition model for prediction of admission scores of colleges and universities in Chinese college entrance examination", PLoS ONE, Volume 17, Issue 10, 2022, Pages 1-17. https://doi.org/10.1371/journal.pone.0274221
[31] Natalia Gil Canto, Marcelo Albuquerque de Oliveira, Gabriela de Mattos Veroneze, "Supervised Learning Applied to Graduation Forecast of Industrial Engineering Students", European Journal of Educational Research, Volume 11, Issue 1, 2022, Pages 325-337. https://doi.org/10.12973/eu-jer.11.1.325
[32] Youdao Wang and Yifan Zhao, "Three-stage feature selection approach for deep learning-based RUL prediction methods", Quality and Reliability Engineering International, Volume 39, Issue 4, 2023, Pages 1223-1247. https://doi.org/10.1002/qre.3288
[33] Rung-Ching Chen, Christine Dewi, Su-Wen Huang &RezzyEkoCaraka, "Selecting critical features for data classification based on machine learning methods", Journal of Big Data, Springer, Volume 7, 2020, Pages 1-26.
https://doi.org/10.1186/s40537-020-00327-4
[34] DaHuang,ZhaoguoLiu and Dan Wu, "Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction", Applied Sciences, Volume 14, Issue 1, 2024, Pages 1-16. https://doi.org/10.3390/app14010040
[35] Narayan Prasad Dahal and Subarna Shakya, "An Analysis of Prediction of Students' Results Using Deep Learning", Computing Open, Volume 1, 2023, Pages 1-28. https://doi.org/10.1142/S2972370123500010