IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 1 July 2024, accepted 18 July 2024, date of publication 29 July 2024, date of current version 6 August 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3434644

RESEARCH ARTICLE

A Hybrid Deep Learning Model to Predict High-Risk Students in Virtual Learning Environments

JAFAR ALI IBRAHIM SYED MASOOD[1], N. S. KALYAN CHAKRAVARTHY[2], DAVID ASIRVATHAM[3], MOHSEN MARJANI[4], DALIA ABDULKAREEM SHAFIQ[5], AND SRINU NIDAMANURI[2]

[1]Department of IoT, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
[2]Center for Data Science, School of Computer Science and Engineering, QIS College of Engineering and Technology, Ongole 523272, India
[3]Faculty of Innovation and Technology, Taylor's University, Subang Jaya 47500, Malaysia
[4]Digital Innovation and Smart Society Impact Laboratory, Faculty of Innovation and Technology, School of Computer Science, Taylor's University, Subang Jaya 47500, Malaysia
[5]Digital Innovation and Smart Society Impact Laboratory, Faculty of Innovation and Technology, Taylor's University, Subang Jaya 47500, Malaysia

Corresponding authors: Jafar Ali Ibrahim Syed Masood (jafarali.s@vit.ac.in) and Mohsen Marjani (Mohsen.Marjani@taylors.edu.my)

This work was supported by Taylor's University and Iverson under Project IVERSON/2018/SOCIT/001.

ABSTRACT Online learning has accelerated with the development of the Internet and communication technology. The widely accessible open online courses are delivered using digital environments that allow students to participate at speed and location. Virtual learning environments (VLEs) have developed quickly in recent years, giving students access to high-quality digital resources. Online learning environments have numerous benefits but drawbacks, including poor engagement, high dropout rates, low engagement, and self-regulated behavior, making students define their aims. Forecasting failed students in a VLE can help organizations and teachers improve their pedagogical practices and make data-driven decisions. This work proposes a Hybrid Deep Learning (HDL) approach to predict students' performance utilizing ECNN (Enhanced Convolution Neural Networks) Resnet model-based classification algorithms. The HDL approach is evaluated using the OULAD (Open University Learning Analytics Dataset), which provides a comprehensive and reliable assessment of the model's performance. The hybrid DLT approaches, demonstrating superiority, exhibited greater prediction accuracy than the existing classifiers. Additionally, the models' accuracy increases by about 95.67%, higher than other approaches are DFFNN model (93.9%) and MLP model (71.41%).

INDEX TERMS Academic performance of students, virtual learning environments (VLE), min-max normalization, hybrid deep learning framework, butterfly optimization, enhanced convolutional neural network (ECNN), Resnet model.

I. INTRODUCTION

Student accomplishments are vital in higher education as they are quality measures of a university's academic success record [1]. Many higher education institutions have established that high-quality education can change students' mental abilities, awareness, and knowledge levels. Teachers seek strategies to increase student accomplishments and enhance teaching process effectiveness continuously [2]. Recent technological advancements and DMTs (Data Mining Techniques) allow instructors to examine and analyze online databases for patterns representing student behaviors and learning. Despite the importance of student performance to the learning process, it is a complicated phenomenon impacted by various elements, including the teaching environment and personal study habits. There are several definitions of student performance [3]. And analyses of student successes in their co-curricular activities for learning

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Li.

evaluations. However, most studies assert graduation as a gauge of students' development.

A. DATA MINING AND MACHINE LEARNING

Traditional DMTs' applications to address issues in education are called EDM (Educational Data Mining), and educational data includes student information, academic performances, and test scores. Reference [4], class participation and student query frequencies are analyzed. Recently, EDM has become a powerful technique for forecasting academic successes, uncovering hidden patterns in educational data, and improving learning and teaching environments [5]. EDM applications have given learning analytics a new perspective, including aspects of student data acquisition, understanding learning environments by inspections and analyses, and determining the ideal performances of students and teachers [6]. Data about students and their environments, including institutions' employment of novel strategies, are gathered, assessed, and reported using learning analytics to comprehend educational procedural changes better.

Some research studies have employed educational data mining to forecast academic achievement; nevertheless, most of these studies have concentrated on traditional high schools and universities with little attention to college education [7]. However, the CGPA (cumulative grade point average) is the primary emphasis of the dataset utilized in earlier studies, along with additional demographic information, ultimately making CGPA feature with maximum information gains. GPA (Grade Point Aggregates) of the courses that students took must be considered in the study since the CGPA combines elements of numerous GPAs across semesters and levels [8]. This work intends to uncover markers for high performances amongst teachers-in-training and predict future academic successes using academic datasets. Risk levels are assigned to existing academic standings for quick monitoring and evaluations to assure better outcomes and boost learning metrics before graduation.

MLTs (Machine Learning Techniques) can uncover key patterns in decisions and policies, considerably improving research activities in the education business [9]. The weakness in educational standards has been attributed to several causes to reverse the bad trend. Hence, further studies are required to improve elementary education, which helps to stop bad trends. Since most teachers at primary schools are trained exclusively in colleges of education, which primarily produce instructors meant for the education industry, therefore may need help understanding the behavior of students and the exact reasons for failing their courses. Using DLT (deep learning approaches)-based mining, focused studies enable the management and evaluation of educational data obtained from multiple sources [10]. Many statistical techniques, data mining, visualization, and machine learning tools are used to analyze educational data. The purpose of the research analytics produced from academic data is to examine information received from institutional databases. Frameworks for learning management evaluate data, enhance teaching techniques, and change the environment in which data is received. Forecasting failed students in a VLE can help organizations and teachers improve their pedagogical practices and make data-driven decisions [11]. This study proposes an HDL (hybrid deep learning) framework to forecast students' performance better using ECNN and Resnet model-based classification algorithms.

B. RESEARCH HIGHLIGHT & CONTRIBUTION

Major contributions to the study can be summarized as follows:

(1) ECNN (Enhanced Convolution Neural Networks) and Resnet model-based classification algorithms.
(2) Butterfly optimization-based feature selection approaches to choose the top features from the dataset connected to students' performance.
(3) The performance of the weight parameter used proposed ECNN algorithm.
(4) The performance evaluation metrics are precision, recall, f-measure, and accuracy.

C. NOVELTY OF THE STUDY

Hybrid Deep Learning (HDL) utilizes ECNN (Enhanced Convolution Neural Networks) Resnet model-based classification algorithms to predict students' performance. Butterfly optimization-based feature selection approaches choose top features from the dataset connected to students' performance. Finally, the proposed HDL achieves higher accuracy in predicting students' performance.

D. RESEARCH QUESTIONS

(1) Why is butterfly optimization better than other optimization methods?
   a. Butterfly optimization is interesting in solving real-life and engineering optimization problems due to its simplicity, efficiency, and robustness.
   b. On the other hand, the symbiosis organisms search algorithm has proved its efficiency in solving complex real-life optimization problems.
(2) Why is the proposed work classification algorithm better than existing algorithms?
   a. Other classification algorithms are not suitable for larger datasets and time-consuming, expensive costs for real-world environments.
(3) Has the proposed work achieved higher accuracy?
   a. Yes, the proposed work achieved higher accuracy than other existing approaches.

The remainder of the research is organized as follows: section II examines current methods for estimating student performance using various MLTs and the current research gaps. Section III outlines the methodology's recommended approach. The findings and discussion are presented in section IV. Section V covers the conclusion, limitations, and future work.

II. LITERATURE REVIEW

This section reviews recent techniques for predicting students' academic performance using MLTs and DLTs.

WVC(Weighted voting classifier), 10-fold cross-validation, and five additional MLTs were joined to create a hybrid method by Ayienda et al., [12] Including SVM (support vector machine), MLPs (multi-layer perceptrons), LRs (logistic regressions), KNNs (k-nearest neighbors), and NBs (Naive Bayes). The study obtained a student grade prediction dataset from Kaggle and examined their proposed model. The performance metrics used for evaluations included obtained accuracies, precision/recall values with computed f1-scores, and AUC (area under the curve), where the study obtained 97.6% accuracy.

Compared to classifiers, Kalyani et al. [13] Predictions based on neural networks yield higher outcomes. The information utilized to predict successes includes hours counts spent studying, their engagements in academics, and other contributing factors. These variables can be extremely important in determining the success of students. CNN (Convolution Neural Networks) is crucial to forecast students' performances. Linear SVM as a gauge of student achievements was proposed by Naicker et al. [14]. Their quantitative research utilized experimental research, with their trials built based on a dataset of 1,000 student records using feature selection. Linear SVMs beat ten categorical MLTs in the baselines for forecasting student progress. The study's outcomes found food as a major hindering factor that influenced reading and writing performances and impacts on mathematics due to race and gender.

Wang et al. [15] They presented SPC (Sequence-based Performance Classifier), a two-stage classification system that combined sequence encoders with traditional data mining classifiers. To more clearly identify sequential features from behaviors, an attention-based HRNN (Hybrid Recurrent Neural Networks) to encode students' campus behaviors was proposed, where more weights were assigned to students' prior activities. Next, they incorporated these newly learned characteristics into traditional SVM algorithms to undertake student performance predictions, leading to the creation of the SPC model. In-depth tests were carried out using the actual student card dataset. The experimental findings show that their suggested strategy was superior regarding recall and accuracy values.

Neha et al. [16] They developed mathematical methods to examine students' academic achievements based on both internal/ external indicators. Many predictive characteristics were considered, including evaluations of student performances by modeling efficient templates for student performance assessments. Their suggested approach employed DNN (Deep Neural Networks) to evaluate predictive variables for student performances, where their results showed that, when the suggested model was contrasted with conventional models, their accuracy levels were much higher.

Begum and Padmannavar [17] Propose an ensemble classifier-based student-predicting model. Data is first preprocessed, and potential redundancies are detected and eliminated to determine higher correlations between qualities. The filtered attribute is learned and evaluated using Boosting, Bagging, and Random subspace classifiers. GA is applied to three classifiers to increase the prediction model's accuracy. A method called GA (Genetic Algorithm) is used to identify the best solutions to search issues to increase the likelihood that the issue will be resolved. The optimization process entails choosing the optimal choice from the alternatives to obtain the intended result. To enhance efficiency and avoid error, the selection is made. An extremely strong positive correlation between the admission qualities led to an investigation of their correlation to determine whether redundancies could exist between them. When Portuguese and mathematical data were evaluated, the classifier's accuracy increased by 3% and 11%, respectively.

Alsariera et al. [18] They proposed the usage of MLTs to help predict student successes. They used 6 MLTs: DTs (decision trees), ANNs (artificial neural networks), SVM, KNNs, LRs, and NBs. Their outcomes in experiments showed ANN's superiority in performance metrics. Furthermore, variables signifying academics, demographics, internal assessments, and personal were major inputs (i.e., predictive features) for predicting student performances.

Damuluri et al. proposed an SVM, one of the most effective classification algorithms currently in use [19]. It can project students' final scores and identify those likely to fail the course so that teachers may provide them with the necessary support. After comparing projected grades to actual grades, our algorithm can predict with 70% accuracy and is the first to successfully determine students' academic results using just online LMS (Learning Management System) data.

Intelligent classification approaches were created by Raut and Nichat [20]. To evaluate performances based on knowledge levels and suggest particular demands for study improvements, such as supporting students through their learning process and making quick decisions to reduce academic risk and desertion. Finally, some suggestions and ideas for the performance's future evolution are presented.

Amrieh et al. [21] provided new data attributes/properties on student behavioral characteristics related to learners' levels of interactions with e-learning management systems. Models for predicting student successes based on DMTs where several classifiers, including ANNs, NBs, and DTs, were used to evaluate the effectiveness of students' predictions. Ensemble approaches were additionally used to increase the performances of these classifiers. The study discovered substantial links between students' behaviors and their academic development. When certain characteristics were removed, the study's suggested model enhanced behavioral aspect prediction accuracy by 22.1%. The usage of ensemble methods increased the accuracy by 25.8%. The model's accuracy, when tested on fresh students, accuracy

TABLE 1. Comparative analysis of the existing approaches.

#	Ref.	Author & Publication year	Approaches used	Results/ Merits	Demerits
1.	[17]	(Begum & Padmannavar, **2022**)	Genetically Optimized Feature Selection with multiclass classification	The proposed methods for Mathematics and Portuguese datasets outperform Relief-Fand Budget Tree-Random Forest with an improvement of 9% accuracy. At the same time, the GA-SVM-based method is 72.3% and 75.9 % in mathematics and Portuguese datasets, respectively, which is very low compared to the proposed method.	The SVM method is the most appropriate for only the unbalanced class distribution problem.
2.	[16]	(Neha et al., **2021**)	Deep Neural Network (DNN) in the process of considering the predictive variables and evaluating student performance using the variables.	It provided 96.24% accuracy in the prediction	Deep Neural networks are complex and require a lot of data to train them.
3.	[14]	(Naicker et al., **2020**)	Linear support vector machines for prediction of student performance in school-based education	It provided 97% accuracy in the prediction.	LSVM could perform better when the data set has more noise, i.e., overlapping target classes.
4.	[13]	(Kalyani et al.,**2020**)	Convolutional Neural Networks are used to predict the performance	It provided 93.81% accuracy, 94.15% precision, 95.13% recall, and 94.64% F1 score.	CNNs typically require large datasets to train effectively. This is because they learn to recognize patterns in data by analyzing many examples of those patterns. If the dataset is too small, the CNN may not be able to learn the patterns effectively and may perform poorly on new data.
5.	[20]	(Raut & Nichat, **2017**)	Students' performance prediction using decision tree	The Decision Tree Model likewise revealed that for the student to pass the Data Structures and Algorithms subject, they should have a grade higher than 66.12% in Midterms and a grade higher than 72.30% in Finals. The Finals attribute serves as the highest indicator that can affect the student.	One of the limitations of decision trees is that they could be more stable compared to other decision predictors. A small change in the data can result in a major change in the structure of the decision tree, which can convey a different result from what users will get in a normal event.

scores were above 80%. The outcome demonstrated the validity of the suggested schema.

To predict academic success, Siddique et al. [22] Built a powerful classification model encompassing multiple classifiers, MLPs, PART, and J48 and ensembles: BAG (Bagging), MB (MultiBoost), and Voting. The study generated nine more models by joining singular or ensembled classifiers for enhanced accuracy. The study's top performer was multi-boost with MLPs, which obtained high values for accuracy (98.7%), precision (98.6%), recall, and F-scores. Recent advancements in Generative Pre-Trained Transformers (GPT) have demonstrated significant improvements in predicting dynamic systems. Studies such as those by [Author1 et al., Neural Netw. 2022, 146, 272-289] and [Author2 et al., Ind. Eng. Chem. Res. 2023, 62, 37, 15278–15289] have shown that GPT models using an encoder-decoder architecture can effectively capture relevant information from input data to predict final system dynamics. These studies indicate that GPT models can achieve higher accuracy than traditional CNNs and other older transformer models, particularly in time-dependent data predictions. While our proposed Hybrid Deep Learning (HDL) approach using ECNN and ResNet models has shown significant accuracy in predicting students' performance, recent studies suggest that Generative Pre-Trained Transformers (GPT) might offer even better performance for time-dependent data. GPT models, which utilize an encoder-decoder architecture, can capture complex patterns and dynamics in the data more effectively. In future studies, Incorporating GPT models could enhance prediction accuracy and provide deeper insights into students' learning behaviors.

A. RESEARCH GAPS

The existing approach, the SVM method, is the most appropriate for only the unbalanced class distribution problem. Deep Neural networks (DNN) are complex and require a

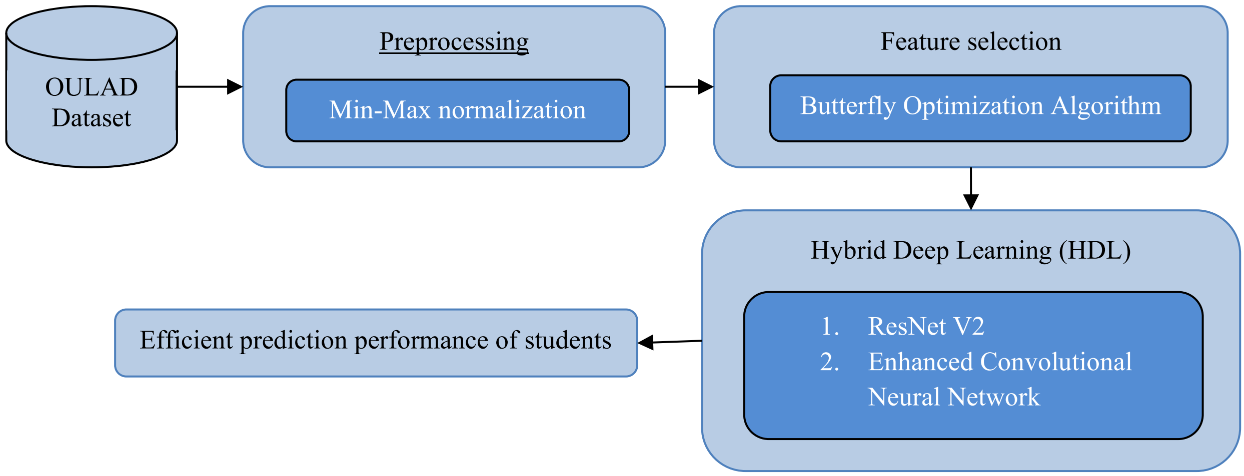

FIGURE 1. The process of student performance prediction using HDL.

lot of data to train them. LSVM could perform better when the data set has more noise, i.e., overlapping target classes. CNNs typically require large datasets to train effectively. If the dataset is too small, the CNN may not be able to learn the patterns effectively and may perform poorly on new data. To solve these issues proposed work introduced ECNN (Enhanced Convolution Neural Networks) and Resnet model-based classification algorithms.

The novelty of the work.

Table 1 shows the Comparative analysis of the existing approaches. Hence, Deep Learning can self-learn and self-adapt, making it extensively studied and successfully used to tackle complex real-world problems.

III. PROPOSED METHODOLOGY

The study technique is thoroughly described in this section. This section provides a detailed explanation of how the data mining technique is implemented. The many phases and trials throughout this research make up the technique. This work proposes the HDL framework to identify the best model that predicts student performances using ECNN and Resnet classifications. Min-max normalization is used to perform the preprocessing at first. Also, this work employed Butterfly optimization-based feature selection approaches to choose the top features from the dataset connected to students' performance. Eventually, the HDL is created to effectively forecast high-risk pupils in a model based on a VLE. The OULAD is used to assess the proposed model.

A. STRENGTHS AND ADVANTAGES OF THE HYBRID DEEP LEARNING (HDL)

The strengths and advantages of the Hybrid Deep Learning (HDL) approach proposed in your study, compared to other relevant studies utilizing different methodologies and datasets:

- **High Prediction Accuracy**

The HDL approach achieved a remarkable prediction accuracy of approximately 95.67%, outperforming several existing classifiers, such as the DFFNN model (93.9%) and MLP model (71.41%). This high accuracy is crucial for effectively identifying high-risk students, enabling timely intervention and support.

- **Incorporation of Advanced Deep Learning Models**

HDL incorporates Enhanced Convolution Neural Networks (ECNN) and Resnet model-based classification algorithms. Integrating these advanced deep learning models allows for more complex feature extraction and learning patterns, enhancing prediction accuracy compared to traditional machine learning approaches.

- **Optimized Feature Selection with Butterfly Optimization:**

The HDL approach utilizes Butterfly optimization-based feature selection, enabling the selection of the most relevant features from the dataset connected to students' performance. This ensures that the predictive model focuses on the most informative features, enhancing efficiency and accuracy in prediction.

- **Efficient Data Preprocessing with Min-Max Normalization:**

The HDL approach initiates with preprocessing using min-max normalization, effectively scaling the dataset to a consistent range. This ensures the features are comparable, leading to more stable and accurate predictions.

- **Addressing Class Imbalance and Noisy Data:**

The HDL approach overcomes challenges related to class imbalance and noisy data often encountered in educational datasets. Techniques within the HDL approach help manage these issues, ensuring the model's robustness and reliability in predicting student performance.

- **Utilization of Publicly Accessible and Accredited Dataset (OULAD):**

The HDL approach utilizes the Open University Learning Analytics Dataset (OULAD), a publicly accessible and accredited dataset provided by the Open University in the UK. Leveraging this widely accepted dataset enhances the credibility and applicability of the research findings.

- **Enhanced Understanding of Students' Performance:**

HDL allows for a deeper understanding of students' performance by leveraging the power of deep learning. By utilizing ECNN and Resnet models, the approach can uncover complex patterns and relationships in student data, enabling a more insightful analysis of academic performance.

- **Comparison of MLP and DFFNN with our proposed HDL Model:**

We select Multilayer Perceptrons (MLP) and Deep Feedforward Neural Networks (DFFNN) for comparison with our Hybrid Deep Learning (HDL) model for the following reasons.

1. ***Baseline Comparison:*** MLPs and DFFNNs are foundational architectures in neural networks. Comparing our HDL model with these well-established networks serves as a baseline test. It allows us to demonstrate the advancements and improvements our HDL model offers over more traditional approaches familiar to the research community.
2. ***Model Complexity:*** MLPs and DFFNNs represent a simpler class of neural networks. Comparing them with the HDL model can highlight the benefits of incorporating more complex architectures (like CNNs and ResNets) in handling the intricacies of data from Virtual Learning Environments.
3. ***Widespread Usage:*** MLPs and DFFNNs are widely used in various applications, making them a logical reference point for many researchers. Demonstrating superior performance or advantages over these common models can strongly position the HDL model as a significant contribution to the field.

   - **Additional Deep Learning Technique**

   We proposed additional deep-learning techniques for comparisons to provide a more comprehensive analysis.

4. ***Recurrent Neural Networks (RNNs):*** Given their ability to handle sequential data, RNNs (and their variants like LSTM and GRU) are particularly relevant in educational data contexts where temporal dynamics (like changes in student engagement over time) are crucial.
5. ***Convolutional Neural Networks (CNNs):*** While our HDL model incorporates CNN elements, a direct comparison with standard CNN architectures can highlight our model's specific enhancements, especially in processing non-image data.
6. ***Transformer Models:*** Recently gaining popularity in various domains, including NLP, transformers can handle large data sequences and might provide interesting insights when applied to VLE data.
7. ***Autoencoders:*** Useful in unsupervised learning scenarios, especially for feature extraction and dimensionality reduction, autoencoders can contrast how different models learn representations in educational datasets.

Overall, the HDL approach demonstrates significant strengths in accuracy, feature selection, data preprocessing, and utilization of advanced deep learning models. These advantages position the HDL approach as a promising method for identifying high-risk students and improving pedagogical practices in virtual learning environments.

B. DATASET DESCRIPTION

In this work, our study utilizes the Open University Learning Analytics Dataset (OULAD), a rich and multifaceted dataset provided by the Open University in the UK. The dataset encompasses a wide array of information about student interactions and performance within Virtual Learning Environments (VLEs), making it an ideal resource for our research on predictive analytics in educational settings. (https://analyse.kmi.open.ac.uk/open_dataset) [23]. Key identifiers are used to connect tables. The student VLE table stores click stream data (number of clicks) that indicate students' daily activities and VLE interactions. The dataset triplet, known as the student-module presentation, contains the assessment results for each student. A version of the OULAD with seven courses, twenty-two two module presentations with thirty-two thousand five hundred and ninety-three students' information gathered between 2013-14. The dataset found at https://analyse.kmi.open.ac.uk/open dataset, OULAD, is publically accessible and has obtained accreditation from the Open Data Institute. When properly assessed and modeled, http://theodi.org/ OULAD offers an early prediction platform for at-risk students.

Nature of the Data

The dataset comprises detailed records from various courses offered by the Open University. It includes data spread across seven tables, each focusing on student and course information. This multi-dimensional nature of the data provides a comprehensive view of the learning environment, student engagement, and performance.

Dimensions of the Dataset

Volume: The dataset encompasses data from 32,593 students across 22 module presentations and seven distinct courses.

Attributes: Key attributes include demographic information, student registrations, VLE interactions (like clickstream data), assessment results, and outcomes.

Specific Attributes Relevant to Our Study

Student Demographics: Age, educational background, and other personal details that offer insight into the diversity of the learner population.

Course Information: Course details, including module specifications and assessment structures.

VLE Interactions: This includes the number of clicks, representing the extent of student engagement with the online resources.

Assessment Results: Continuous and final assessment scores measuring academic performance.

Final Outcomes: Course completion status is pivotal to classifying students as high-risk or successful.

Alignment with Methodology

Our study leverages the depth and breadth of OULAD to develop a predictive model using Enhanced Convolutional

Neural Networks (ECNN) and ResNet model-based classification algorithms. The varied nature of the data allows us to extract meaningful features and patterns related to student performance and engagement. Specifically, we focus on:

Feature Selection: Using Butterfly optimization-based techniques to select the most relevant features from the dataset, particularly those that correlate strongly with student performance and engagement indicators.

Predictive Modeling: Employing our Hybrid Deep Learning (HDL) framework, we analyze these features to predict high-risk students, aiming to improve pedagogical strategies in VLEs significantly.

Our approach, tailored to the unique characteristics of the OULAD, exemplifies the efficacy of applying advanced deep learning techniques to educational data mining, paving the way for more nuanced and impactful educational interventions.

C. RATIONALE BEHIND METHOD SELECTION

In our research, the selection of Hybrid Deep Learning (HDL), Multilayer Perceptrons (MLP), and Deep Feedforward Neural Networks (DFFNN) was driven by the specific characteristics of our dataset and the nature of our research question. Our study focuses on predicting high-risk students in Virtual Learning Environments (VLEs), which involves analyzing complex, high-dimensional, and non-linear data. The rationale for our method selection is as follows:

- **Complex Data Structures:** Our dataset comprises multifaceted features, including time-series data of student interactions and heterogeneous variables, which require advanced modeling to capture complex patterns and relationships.
- **Non-linearity:** Given the non-linear relationships inherent in educational data, deep learning models like HDL offer significant advantages in their ability to model these complexities effectively.
- **High-Dimensional Data:** Our deep learning approaches are well-suited to handle the high dimensionality of VLE data, enabling the extraction of meaningful insights from large and diverse datasets.

D. FEASIBILITY OF INCORPORATING ADDITIONAL TECHNIQUES

- **Discriminant Analysis:** While effective in classification, this technique may not capture the non-linear and high-dimensional nature of VLE data as effectively as deep learning models.
- **CHAID (Chi-squared Automatic Interaction Detection):** Useful in decision tree modeling, CHAID could be employed for exploratory analysis, offering insights into variable interactions. However, its performance in high-dimensional spaces might be limited compared to HDL.
- **Logistic Regression:** While logistic regression is a robust method for binary classification, its application may be constrained in modeling the complexity and scale of VLE data.

In light of these considerations, while these traditional methods have their merits, they may not fully encompass the data's complexity compared to our chosen deep-learning approaches. Nonetheless, future studies could explore a hybrid approach, combining the strengths of traditional and deep learning methods for a more comprehensive analysis. Also, this can clarify the methodological choices made in our study, ensuring that our approach aligns with the specific demands of VLE data analysis. It also acknowledges the potential of other statistical techniques, paving the way for future research to explore a more diverse methodological landscape.

E. DATA PREPROCESSING

To improve the prediction model's performance efficiencies, all occurrences of nulls or values with noises in OULAD were eliminated or replaced with mean values. Since dates are critical variables in the early predictions, all date occurrences with N/A or nulls and replaced with missing values with the column mean values. Min-max normalizations [24] Were used in these operations.

- **Normalization and Transformation**: Before analysis, the data undergoes min-max normalization and is reshaped to fit the input requirements of our deep learning models. This step ensures comparability across features and prepares the data for effective pattern recognition.
- **Min-Max Normalizatio**

The min-max normalization approach converts the input data into a newly defined range while normalizing the dataset using a linear transformation [24]. The min-max approach preserves the linkages between the original input value and the scaled result. Also, an out-of-bound error occurs when the normalized values depart from the original data range. With the help of this method, extreme input values are guaranteed to fall within a certain range. In our study, we have recognized the need for a uniform scaling approach for the variables identified in the dataset. To address this, we have implemented min-max normalization across all variables. This process involves rescaling the range of features to scale the data within the interval [0, 1]. The formula gives the normalization. A value X0 is transformed by Min-Max normalization into a value Xn that falls within the defined range and is represented by the equation (1)

$$X_n = \frac{X_0 - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where X_n stands for variable X's new value, X_0 represents X's current value, and X_{min} represents the dataset's min. Data point while X_{max} stands for the dataset's max data point. By applying this normalization, we ensure that each feature contributes equally to the analysis, eliminating any bias due to the original scale of the data. This step is particularly crucial in our deep learning models, where variable scales can significantly influence the model's learning process. Consistent scaling across all variables enhances the comparability of different features and improves our predictive models' convergence speed and overall performance.

Practical Application in the Study:

Before Model Training: Apply min-max normalization to the entire dataset before dividing it into training and testing sets. This ensures that the scale of the data does not bias the model and learns from the actual relationships between features.

Maintain Uniformity: Using the same scaling parameters (min and max values) for training and testing datasets is important to maintain consistency in model evaluation.

In our methodology, to evaluate the performance of the proposed Hybrid Deep Learning (HDL) model more comprehensively, we have extended our analysis beyond traditional metrics such as accuracy, precision, recall, and F1-score. We now include advanced information criteria - AIC, SBIC, HQIC, and AICc - pivotal in model selection and validation in statistical contexts. These criteria offer insights into the model's goodness of fit while penalizing for complexity, thus enabling a balanced assessment of model performance.

AIC (Akaike Information Criterion):Measures the relative quality of the model, balancing the model's fit and complexity.

SBIC (Schwarz Bayesian Information Criterion): Similar to AIC but with a stronger penalty for models with more parameters.

HQIC (Hannan-Quinn Information Criterion):Offers a compromise between AIC and SBIC, penalizing less for model complexity.

AICc (Corrected Akaike Information Criterion):An adjusted version of AIC, more accurate for smaller sample sizes.

These criteria are particularly beneficial in comparing models with different numbers of parameters, helping to identify the most parsimonious model that sufficiently explains the data.

In this study, we applied the Akaike Information Criterion (AIC), Schwarz Bayesian Information Criterion (SBIC), and Hannan-Quinn Information Criterion (HQIC) as part of our model evaluation process. These criteria were instrumental in selecting the best model for predicting student performance. The results presented in Table 2 demonstrate that models with lower AIC, SBIC, and HQIC values were more effective, thereby providing a more scientific and robust approach to model evaluation."

F. FEATURE SELECTIONS USING BOA (BUTTERFLY OPTIMIZATION ALGORITHM)

Feature Selection Using Butterfly Optimization:

- **BOA Implementation:** We employ the Butterfly Optimization Algorithm (BOA) to select our dataset's most relevant and impactful features. This step is crucial in reducing dimensionality and focusing our models on the most predictive attributes.

BOA, a metaheuristic swarm-based algorithm, was developed to mimic butterflies' foraging and information-sharing behaviors. The following qualities of butterflies are idealized: 1. Butterflies, in general, emit a fragrance to attract other butterflies. 2. Their flights are randomized or in the direction of the strongest scents. 3. The geography of the goal function influences or determines butterfly responses to stimuli and implies BOA is in the last step.

In BOA runs, initialization occurs first, followed by repeated searches, and lastly, the operation is terminated after optimal solutions are determined. Upon initialization, the algorithm creates target functions and their solution spaces. In addition, BOA parameter values are assigned [25]. Following variable definitions, starting populations of butterflies are initiated. Total butterfly counts are static in simulations, and data for butterflies are maintained in fixed memory sizes. The sites are produced randomly in search regions, and fitness and scent values are computed and recorded. This marks the end of the starting steps; the algorithm begins iterating, and fake butterfly searches are carried out.

The second step, sometimes known as the iteration phase, entails several iterations of the procedure. Every butterfly in the solution space travels to a new place after each iteration, at which point their fitness values are computed. The initial step is to identify the fitness values for each butterfly over the solution space. Using electricity, these butterflies will exude smell where they are placed (2). There are two main search phases, namely, global and local. Butterflies move closer to ideal butterflies/solutions, g, during global searches. G can be represented by equation (3)

$$onex_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \tag{2}$$

here, ${}_i^t$ represents the butterfly's answer. The vector xi represents iteration number t, and the symbol g indicates that the top solution was just obtained. Fi represents ith butterfly's scent, while r is the chosen integer value in the range [0, 1]. In local search phases, they are:

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \tag{3}$$

where x_j^t and x_k^t Are butterflies from the solution space jth and kth, equation (3) turns into a local random walk if x_j^t and x_k^t Re members of the same swarm where r is in value ranges between [0, 1]. Both locally and globally, butterflies might search for food and a partner. When physical closeness and numerous other circumstances, such as rainfalls, strong winds, etc., are considered, a butterfly's efforts to find a partner or food may constitute a major portion of its overall activity. BOA uses p-switch probabilities to shift from common global searches to intense local searches. Algorithm 1 explains the BOA pseudocode.

G. ELABORATION ON METHODOLOGY EFFECTIVENESS

- **Feature Selection with Butterfly Optimizatio**

Enhanced Relevance and Accuracy: Using the Butterfly Optimization Algorithm (BOA) in feature selection helps identify the most relevant features from a large dataset. By focusing on

Algorithm 1 Pseudocode of BOA

Input: Objective function f (X), Xi = (X1, X2, …. Xdim),
dim= dimensions count
Maxiter- max. Iterations count
Bf: butterfly counts in the ecosystem

1: **Initialization:**
2: Set G=0 as the generation/iteration number initialization;
3: Create a population of n butterflies starting at i=1, 2,…, n.
4: f determines Stimulus Intensity I at Xi (Xi)
5: Define switch probabilities p, power exponents a, and sensor modalities c.
6: Compute the scent **for each** butterfly in the population if the stopping requirements are unsatisfied.
7: Choose the ideal Bf
8: Generate randomized numbers r in the range [0, 1] for butterflies Xi in populations.
9: **If** $\leq p$ **then**
10: Using equation (2), head in the direction of the ideal butterfly solution
11: Select a butterfly at random (i $\neq$ j);
12: Using equation (3), obtain the mutual relationships vector (Mutual_Vector)
13: Using Equations (4) and (5), update butterflies based on their mutual collaboration partnerships approaches;
14: Compute fitness values of new Butterflies;
15: **Else**
16: **Update** the new position
17: **End for**
18: **Update** the best value
19: **End**
20: **Output:** Best Butterflies with min. Fitness (optimized features)

the most informative attributes, BOA improves the model's accuracy, eliminating noise and irrelevant data that could otherwise lead to overfitting or underperformance.

Optimization Efficiency: BOA mimics the foraging behavior of butterflies, making it a nature-inspired algorithm that is efficient in exploring the search space. This leads to a more effective identification of optimal features than traditional methods like manual selection or basic statistical techniques.

- **Deep Learning Techniques: ECNN and ResNe**

Advanced Pattern Recognition: Enhanced Convolutional Neural Networks (ECNN) and ResNet models can identify complex patterns in data. These deep learning architectures are particularly effective in handling large and intricate datasets, as they can learn hierarchies of features.

Handling Non-linear Relationships: Deep learning models excel at capturing non-linear relationships in data, which traditional machine learning models often miss. This is particularly beneficial in educational data, where a complex interplay of factors can influence student performance.

Resilience to Overfitting: The incorporation of ResNet, with its skip connections, helps mitigate the vanishing gradient problem, allowing the model to be deep without the risk of overfitting. This leads to improved performance on unseen data.

H. ANALYSIS OF TIME EFFICIENCY

- Compared to Traditional Approaches

Training Time: While deep learning models, particularly those with complex architectures like ECNN and ResNet, tend to have longer training times than simpler models, they often converge to a better solution. Advanced hardware (like GPUs) can significantly reduce this training time.

Inference Time: Once trained, deep learning models generally have very fast inference time—the time it takes to make predictions on new data—making them suitable for real-time applications.

Scalability with Data Size: Deep learning models scale efficiently with increased data size, unlike some traditional machine learning models that may suffer from a substantial increase in computational cost.

- Practicality for Real-World Applications

Automated Feature Selection: Automated feature selection using BOA reduces the time and expertise required in the data preprocessing stage, which is often time-consuming in traditional approaches.

Generalization Capability: Deep learning models' ability to generalize well to new, unseen data makes them highly practical for dynamic real-world scenarios, where data patterns may change over time.

Maintenance and Updating: While the initial training might be resource-intensive, deep learning models can be updated incrementally with new data, making them time-efficient in an ongoing operational context.

I. ENHANCED UNDERSTANDING OF STUDENT DATA THROUGH CNNs

1. Advanced Pattern Recognition in VLE Data: Convolutional Neural Networks (CNNs) are exceptionally suited for analyzing the rich and multifaceted data generated in Virtual Learning Environments (VLEs). Unlike traditional flat-structured models, CNNs are adept at recognizing complex patterns within high-dimensional spaces. VLEs generate many data points – from engagement metrics to performance indicators. CNNs, with their deep-layered structure, can decipher intricate interactions within this data. Lower layers of the network capture basic interactions, such as login frequencies, while deeper layers synthesize these into more abstract representations, like learning behaviors or styles. This hierarchical feature learning mirrors the layered complexity of VLE data, where basic metrics intertwine to form deeper educational insights.

2. Uncovering Subtle and Complex Interactions: CNNs use convolutional filters to automatically and adaptively focus on significant features, highlighting subtle but crucial patterns in student data. This capacity to dynamically identify predictive features is a marked advancement over traditional models that rely on predefined feature sets. For instance, CNNs can detect changes in student engagement levels and how they correlate with academic performance, uncovering non-linear and higher-order interactions vital to understanding student behavior in VLEs. This deep, nuanced analysis enables CNNs to uncover subtle behavioral patterns and learning trajectories that more traditional analytical methods might overlook.

3. Robustness to Variability in Educational Data: CNNs are inherently robust to the common challenges in VLE datasets, such as noisy, incomplete, or inconsistent data. They excel at filtering out irrelevant information, thus enhancing the precision of predictions. This robustness is particularly beneficial in educational settings where data can vary widely due to the diversity in student backgrounds, behaviors, and learning paths. By managing these data inconsistencies effectively, CNNs ensure that the subtle yet significant patterns within the data are not obscured, leading to more reliable insights into student learning processes.

4. Enhanced Predictive Accuracy and Educational Insights: The application of CNNs in VLE data analysis leads to more accurate predictions of student outcomes. By capturing nuanced patterns and complex relationships within the data, CNNs offer a more precise understanding of student behaviors and learning outcomes. This enhanced predictive accuracy is not just a technical achievement; it has practical implications for education. Educators can leverage these insights for more targeted interventions, improving student engagement and success in digital learning environments. The adaptability and scalability of CNNs also mean that they can evolve with the ever-increasing volume and complexity of educational data, maintaining their effectiveness and relevance in diverse learning contexts.

By summarizing the above context, integrating CNNs in analyzing VLE data represents a significant step forward in educational data analytics. By offering a deeper, more comprehensive understanding of student behaviors and learning processes, CNNs enable educators to make informed, data-driven decisions, ultimately enhancing the quality and effectiveness of education in digital environments.

Detailed Explanation of CNN Application to Tabular Data: Traditional Usage: CNNs are predominantly used in image processing tasks due to their ability to detect spatial hierarchies in data. Each layer captures a level of abstraction, making them powerful for image recognition.

1. **Adapting CNNs for Tabular Data**
   - ❖ **Tabular Data Characteristics:** Unlike image data, tabular data is structured with rows and columns, where each column represents a variable or feature, and each row represents a data record. This data doesn't have the spatial relationships that images do.
   - ❖ **Data Preprocessing:** To apply CNNs effectively to tabular data, the data needs to be preprocessed and transformed. This includes normalization and potentially reshaping the data to create a pseudo-image structure.
2. **CNN Architecture for Tabular Data**
   - ❖ **Input Layer Adaptation:** The input layer of the CNN is modified to accept one-dimensional data (representing the tabular format). This involves reshaping the data into a format suitable for convolutional operations.
   - ❖ **Convolutional Layers:** Convolutional layers detect patterns and relationships between different features in the data. The filters in these layers slide across data rows (or reshaped structures), capturing dependencies between features.
   - ❖ **Pooling Layers:** These layers reduce the dimensionality of the data, helping to prevent overfitting and reducing computational load.
   - ❖ **Fully Connected Layers:** After convolution and pooling layers, the network uses fully connected layers to interpret the feature maps and make predictions.
3. **Theoretical Justification**
   - ❖ **Feature Interaction Learning:** CNNs in this application learn interactions between different features in the tabular data. This is akin to how they learn spatial hierarchies in images.
   - ❖ **Dimensionality Reduction:** Through convolution and pooling, CNNs reduce the number of parameters, helping to manage the curse of dimensionality often associated with high-feature datasets.
4. **Practical Considerations**
   - ❖ **Hyperparameter Tuning:** The choice of kernel size, number of filters, and architecture depth must be tuned based on the specific dataset.
   - ❖ **Data Representation:** Careful consideration is needed in how the data is reshaped or represented to ensure meaningful convolution operations.

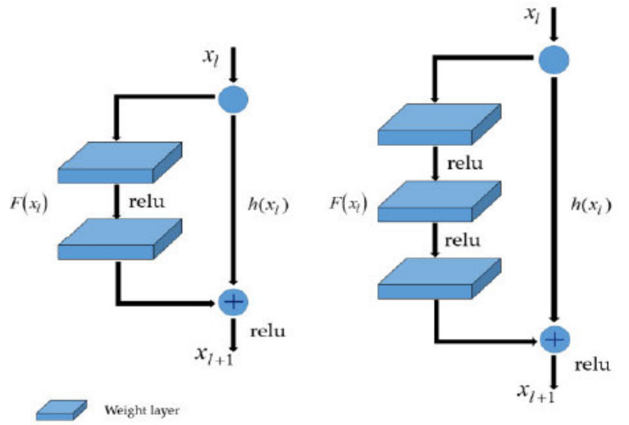

FIGURE 2. Structure of Two/Three layered residual networks.

J. CLASSIFICATION USING HDL MODEL

This study uses the HDL model to categorize student performance prediction. The ResNet V2 and improved CNN are combined here for effective prediction outcomes.

- **ResNet V**

CNN ResNet-v2 was constructed using training data from the input database containing over a million pictures. With the 164-layer network, photos may be classified into 1000 unique item categories, including keyboard, mouse, pencil, and various animal categories [26]. As a consequence, the networks now have detailed feature representations of attributes.

Figure 2 displays the remaining network structure. The ability to send signals forward and backward from different units and layers to any layer speeds up network training and parameter tweaking. The ResNet is composed of two branch networks and a backbone network.

Dual-layered networks train on high-resolution and target attributes to transform them as linked attributes with differing resolutions, while the backbone network recognizes objects in an attribute. The residual convolution networks' dimension must be increased or decreased using 1∗1 convolutions since the feature map counts for xl and may change. Residual operations can be represented as:

$$F\left(x_l\right) = w * x_l + \alpha \tag{4}$$

$$y_l = R\left(F\right) + h(x_l) \tag{5}$$

$$x_l + 1 = R(y_l) \tag{6}$$

x_l implies inputs in these equations; w, the e weights; the offset; y_l represents the sum of two branches; R represents Relu functions; $F(x_l)$ stands for convolution operations; $h(x_l)$ implies simple transformations of inputs while (x_l+1) represents residual module's final outputs. Relu is an activation function that aids in the spread of the ladder and keeps it from diverging to keep steps from being severely attenuated behind multi-layer convolutions, as shown in equation (7).

$$R\left(x\right) = \max(0, x) \tag{7}$$

When $x > 0$, $R(x) = x$, and its lead is 1; when $x \; is \leq 0$, $R(x) = 0$, with a lead of 0. Enter the value x and the threshold 0 in the forward computation to get the outcome. The gradient in the backward computation is either 1 or 0. In other words, there is little to no gradient drop. Relu's ease of computation and lower degree of gradient reduction than other functions like Tanh and the Sigmoid are advantageous for deepening networks. The remaining network layers can be similarly mapped when the model reaches specified performance saturations, which speeds up and facilitates training network convergence. The residual function is denoted as F, where Xi and Xn are the inputs of the ith and nth residual units. The gradient will never approach 0, regardless of the network layers' depth. The formula for learning characteristics from shallow i layers to deep n layers is:

$$\frac{\partial X_n}{\partial X_i} = \frac{\partial X_i + F(\partial X_i, \omega i, \alpha i)}{\partial X_i} = 1 + \frac{\partial F(X_n, \omega n, \alpha n)}{\partial X_n} \tag{8}$$

Residual networks have three layers that first use 1∗1 convolutions to reduce dimensionalities, followed by 3∗3 convolutions. Compared to two-layer residual network units, tri-layered residual networks have network parameters 17.35 times less. ResNet has shown to be efficient, but one big drawback is that it sometimes takes weeks to train a deeper network. Hence, this study reported enhanced convolutional neural networks for efficient ResNet model training.

- **ECNN**

A specific kind of FFNN (feed-forward neural network) that uses pooling, convolution, and ReLU is known as CNN. A typical CNN comprises the Feed-forward Neural Network layers, including fully connected, convolution, and pooling layers. Each connection between neurons in one layer and those in the layer above often acts as a network parameter in conventional ANNs. This might lead to very parameter counts. CNNs employ local connections between neurons rather than interconnected layers, which implies that neurons get connected only to subsequent layer neurons. Hence, the total counts of network parameters may significantly decline.

Moreover, weights are used in connections between local receptive fields and neurons. Kernels are a collective name for these sets of weights. Neurons connected to local receptive fields maintain computational results between local receptive fields and neurons in matrices called activation maps. CNNs share weights resulting in generations of different activation maps from different kernels, and hyper-parameters are used for changing counts of kernels. Irrespective of connections between neurons, weight counts correlate with the sizes of local receptive fields or kernels.

A convolution layer is required to build CNN to convert inputs into representations of more abstract levels [28]. The convolution layer uses a local connection rather than full connectivity to execute calculations between the input and the hidden neurons. By moving at least one kernel over inputs, convolution layers perform convolution operations between inputs and kernels. The activation maps contain findings, which are the convolution layer's outputs. It is important to remember that activation maps could include features that different kernels have retrieved. Kernels can act like feature extractors, sending their weights to neurons. Several spatial parameters must be provided for convolution methods to produce activation maps of sizes. Among the necessary qualities that are

(1) **The kernels' size (N):** Every kernel contains a window size, also known as a receptive field. Kernels convolve portions of inputs that match window sizes to generate their activation maps.
(2) **Stride (S):** The number of pixels the kernel will move to the next location depends on the value of this parameter during convolutions around input volumes and subsequent movements of pixels before chosen borders are reached if it is set to 1. Thus, strides may be employed to accomplish end outcomes since the sizes of activation maps are reduced as the stride grows.
(3) **Zero-padding (P):** This option determines zero counts that need to be added to inputs and helps maintain input dimensions.

Local connections and shared weights significantly reduce the network's parameters. Kernel's dimensions are 4 × 4 x 3 when convolution layers and kernels count are equal to 2, and kernels have 4 pixels as local receptive fields. The input volume's depth will match the kernel's three depths. Because the weights of each kernel are the same for each of the layer's 100 neurons, there will only be $4 \times 4 \times 3 \times 2 = 96$ parameters in this layer. This considers kernel counts and not the counts of neurons in layers, as well as the strength of the neighboring connections. Shared weights, local connections, and reduced parameter counts are critical for effective picture processing.

Furthermore, because local values in pictures are closely connected and generated local values are frequently location-invariant, local convolution processes in an attribute yield specific attributes. A kernel with the same weights may thus extract patterns from every local area of the picture, while other kernels can extract patterns of different sorts from the attribute. It is usual practice to apply the outcomes of the convolution processes between the input and kernel to a non-linear activation function (for example, ReLu, tanh, sigmoid). These values are contained in the activation maps and transmitted to the following network layer.

- **Back Propagation Algorithm**

The CNN algorithm's two fundamental steps are convolution and sampling. The stages in the convolution process are using a trainable filter F_x, deconvolution of the input attribute (the first stage; the input of the convolution that follows is the feature attribute of each layer, especially the Feature Map), the addition of a bias bx, and formation of the convolution layer C_x. a sampling strategy in which one pixel is created by combining n pixels from each neighborhood. A scalar function then weights this pixel to add bias ($b_x + 1$) and produce a tiny n times feature map ($S_x + 1$). CNNs employ local receptive fields, weight sharing, and time/space-based sub-samples to extract features and minimize training parameter sizes.

$$O_{x,y}^{(l,k)} = tanh \sum_{t=0}^{f-1} \sum_{r=0}^{K_h} \sum_{c=0}^{K_w} W_{(r,c)}^{(k,t)} O_{(x+r,x+c)}^{(l-1,t)} + Bias^{(l,k)} \quad (9)$$

Among them, f stands for convolution core counts in feature patterns. Outputs of neurons of rows x, columns y in the l th sub-sample layers and k^{th} feature patterns:

$$O_{x,y}^{(l,k)} = \tanh (W^k \sum\nolimits_{r=0}^{S_h} \sum\nolimits_{c=0}^{S_w} O_{(x \times S_h + r, y \times S_w + c)}^{(l-1,t)} + Bias^{(l,k)} \quad (10)$$

The outputs of j th neurons in l th hidden layers H:

$$O_{(i,j)} = \tanh (W^k \sum\nolimits_{k=0}^{S-1} \sum\nolimits_{x=0}^{S_h} \sum\nolimits_{y=0}^{S_w} W_{(x,y)}^{(j,k)} O_{(x,y)}^{(l-1,t)} + Bias^{(l,k)} \quad (11)$$

Among them, s stands for feature pattern counts in sample layers. Outputs of i th neuron l th output layer F.

$$O_{(i,j)} = \tanh(\sum\nolimits_{j=0}^{H} O_{(l-1,j)} W_{(i,j)}^{l} + Bias^{(l,i)} \quad (12)$$

Between convolution layers and following layers, pooling layers are frequently included. Pooling layers use predetermined pooling approaches to conserve as much data as feasible while reducing input dimensions. Pooling layers can also give networks spatial invariance, enhancing the generalizability of models. Pooling layers use stride, zero-padding, and pooling window sizes as hyper-parameters for their pooling operations. Like the kernel in a convolution layer, the pooling layer will use the predetermined pooling window size to scan the whole input. For instance, pooling sizes are cut in half using strides of 2, window sizes 2, and 0 zeros-padding.

A fully connected layer is the essential building block of a hidden layer in CNN. In conventional CNN systems, a fully connected layer is typically positioned between the penultimate and output layers to mimic the input data's non-linear correlations more accurately. However, this benefit has lately been questioned due to its numerous elements and the possibility of overfitting. As a result of this endeavor, MPSO (modified particle swarm optimization algorithm) was introduced.

- **PSO (Particle Swarm Optimization) algorithm**

Based on individual bird flock behaviors, PSO solves scalability issues. The algorithm mimics the social behaviors of creatures living in sizable groups that interact and work together. Particle adjusts their and neighbors' movements in PSO in search spaces for rapid movements. A particle swarm in the fundamental PSO comprises 'n' particles [27]. The position of each particle in the D-dimensional space symbolizes a practical solution. Private initiatives and technological solutions move in hyper-dimensional search spaces. A particle's transitions within the swarm are influenced by its perception or knowledge of its neighbors. Just three steps make up the PSO algorithm, which is continued until the situation is resolved. They defy all logic.

(a) Establish the fitness of each particle.
(b) Continue to update the best people and worldwide functions.
(c) Improve the placement and intensity of each particle.

Best-fit Particles are located for swarms. Search space poses ⬚ are identifiable, velocity VY and personal best P_{bt}, Th a minimum value determined by objective function fn is defined for every individual particle k€[1..n] where $n > 1$. This $P_{bt,k}$, is used to measure G_{bt}, The finest scenario, which is the outcome of making comparisons of all the $P_{bt,k}$.

This P_{bt} is entirely predicated on the equation,

$$P_{bt,k} = \begin{cases} P_{bt,k} & if\ f_n(q_k) > P_{bt,k} \\ q_k & if\ f_n(q_k) \leq P_{bt,k} \end{cases} \quad (13)$$

The best global position equation for the calculation G_{bt} is,

$$CloseCloseG_{bt} = \{\min \{P_{bt,}\}\ where\ k \Delta [1 \ldots .]\ where\ n > 1 \quad (14)$$

keep updating velocity,

$$VY_k = s_k vy_k (r) + a_1 b_1 [p_k (r) - -u_k (r)] + a_2 b_2 [g_k (r) - u_k (r)] \quad (15)$$

VY_k is velocity, $a.\,sub\,2\,are\ client-based\ coefficient\ values$s. b_1 and b_2 signify randomized values while the individual solution $g_k (r)$ s unified warm's global best solution variables are denoted by s_k⬚(⬚) y. PSO begins with a set of random solutions and then searches for optimal solutions using more recent generations. After the two best values in each cycle, each particle is updated. Thus far, the first choice has shown to be the most beneficial. The coefficient values of A_1 and A_2 <= 2. Because of the inertia variable, the particle will always travel in the same direction it started.

This has a range of 0.8 to 1.2. When the inertia cost is high, the swarm converges best. Larger numbers allow you to search the whole region. The recognized cognitive variables $b_1 [p_k (r) - u_k (r)]$. This asserts as a particle's memory and controls a return to the search region, where the individual factors are important. Particles move towards the best warm zones that they experience during special interactions. $b_2 [g_k (r) - u_k (r)]$

This research uses an MPSO variant that tries to improve the PSO algorithm's efficiency in discovering superior solutions while retaining its simplicity and speedy convergence. This Gaussian factor is based on including a simple yet effective new operation into the iterative search process to boost the algorithm's capacity for exploring new search space regions that may hold better answers and using intermediate solutions. The proposed alternative is based on the revised PSO version, depending on parameter values.

- **Gaussian factor**

The particles in PSO will assemble at a certain location even though the global optimum has yet to be found since the dimensions of a feature vector are often quite high. The Gaussian factor K was added to PSO to provide the optimum convergence. The velocity formula is presented in Formula (16):

$$v_{id} = K[v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times \left(p_{gd} - x_{id}\right)] \tag{16}$$

This study calculated the Gaussian factor K using the suggested formula. Values c1 and c2 both utilized 2.05, matching Clerc's experiment exactly. Four decimal points of K are set aside here for experimentation. The precise velocity formula is given in formula (17):

$$v_{id} = 0.7298 \times [v_{id} + 2.05 \times rand() \times (p_{id} - x_{id}) + 2.05 \times Rand() \times \left(p_{gd} - x_{id}\right) \tag{17}$$

A particle in PSO must first be detected across a large range to find the most likely position of the best solution. More localized iterations must emerge inside a confined space to locate the ideal place. K should choose a higher value early on and a lower number later. K should then gradually decrease over a longer period until it reaches its minimum. This changing pattern is consistent with the concave function.

The Gaussian factor should select a convex function in the first iteration to avoid premature convergence, allowing the particles to discover the best solution across a large range. It should use a concave function in the late hours to allow for local growth and a progressive movement of the Gaussian factor to its lowest value. It guarantees algorithmic convergence. Formula (18), which depicts the functional Gaussian factor structure based on the cosine function, demonstrates this concept:

$$K = \frac{\cos((\pi / G_{max}) \times T) + 2.5}{4} \tag{18}$$

T is the number of iterations. The shifting curve of value K was visible when G_{max} was set at 40. The K curve turns from a convex function to a concave one over time. Formula (14), which substitutes the value K, becomes formula (15) as a result (19). The following describes formula (19):

$$v_{id} = \left(\frac{\cos((\pi \times T / G_{max})) \times 2.5}{4}\right) \times [v_{id} + 2 \times rand() \times (p_{id} - x_{id}) + 2 \times Rand() \times \left(p_{gd} - x_{id}\right)] \tag{19}$$

The HDL model accurately predicts student performance and provides this information to the course teacher. It is the most effective prediction model for raising student performance.

Implementation of K-Fold Cross-Validation:

1. Choosing K Value:

The first step is determining the number of folds, commonly denoted as 'K.' In practice, K=10 is a widely accepted choice, as it offers a good balance between computational efficiency and the reliability of the results.

2. Dividing the Dataset:

Randomly divide our dataset into K equally (or nearly equally) sized folds or subsets. Each fold should be representative of the whole dataset, ensuring that all variations within the data are present in each fold.

3. Iterative Training and Validation:

Perform K iterations of training and validation. Each iteration uses one-fold as the validation set and the remaining K-1 folds as the training set. This process ensures that every data point is used for both training and validation exactly once.

4. Model Training:

In each iteration, we have trained our Enhanced Convolutional Neural Networks (ECNN) and ResNet models on the training set. Then, the trained model was validated on the validation set reserved for that iteration.

Evaluate the model's performance in each iteration using metrics appropriate for our study, such as accuracy, precision, recall, and F1-score.

5. Aggregating Results:

After completing all K iterations, aggregate the performance metrics from each iteration to get a comprehensive view of the model's performance. Calculate the average of these metrics across all K iterations. This average performance is a more robust estimate of our model's effectiveness, as it is less sensitive to the particularities of a single data split.

6. Reporting:

In our research, we report the average performance metrics and their standard deviations to provide insight into the variability of our model's performance across different folds.

7. Interpretation:

Use the K-fold cross-validation results to interpret our model's robustness. If the model performs consistently across different folds, it generalizes well to new data.

Benefits of K-Fold Cross-Validation in our Study:

- **Robustness:** Provides a more reliable estimate of model performance, especially important for datasets with limited size or variability.

- **Unbiased Evaluation:** The evaluation is less biased since every data point is used for training and validation.
- **Model Tuning:** Helps in identifying the best hyperparameters for our model, enhancing its predictive power.

IV. RESULTS &DISCUSSION

Performances of at-risk students were predicted using criteria related to student demographics, VLE engagement, and assessments at various percentages of the course length. Figure 1 depicts this workflow, which is divided into six phases: the start of the course, 20%, 40%, 60%, 80%, and 100% of the course studied. To mimic OULAD, these algorithms were developed to divide student performance into four categories (students who were unable to complete the course), Pass (students who finished courses with passing grades), Fail (those who did not), and Distinction (those who did not) (completed courses with excellent grades).

The proportion of accurately discovered positive observations to all anticipated positive observations is how precision is defined.

$$Precision = TP/(TP + FP) \tag{20}$$

The proportion of properly detected positive observations to all observations is known as sensitivity or recall.

$$Recall = TP/(TP + FN) \tag{21}$$

The weighted averages of Precision and Recall values are F-measure values, encompassing false positives and negatives.

$$F - measure = 2 * (Recall * Precision)/(Recall + Precision) \tag{22}$$

Accuracy is calculated in terms of positives and negatives as follows:

$$Accuracy = (TP + FP)/(TP + TN + FP + FN) \tag{23}$$

where TP- True Positive, FP- False Positive, TN-True Negative, FN- False Negative.

Figure 3 thoroughly compares the suggested HDL and the current method for classifying student risk levels. The suggested predictive model's outcomes showed that it was effective in early predicting at-risk students. These outcomes can assist VLE administrators/ instructors in developing online learning frameworks that impact choices.

Fig. 4 displays the recall comparison between the proposed student prediction model and the currently used one. The prediction model is therefore seen as a classification problem resulting from the student having either a high or low risk depending on the course duration. As a result, the recommended HDL models were applied to the given job, and the results were reviewed and evaluated.

In Fig. 5, the proposed student prediction model is compared to the current student prediction model using the F-measure. The results demonstrated that the HDL model outperformed the considered machine-learning techniques on

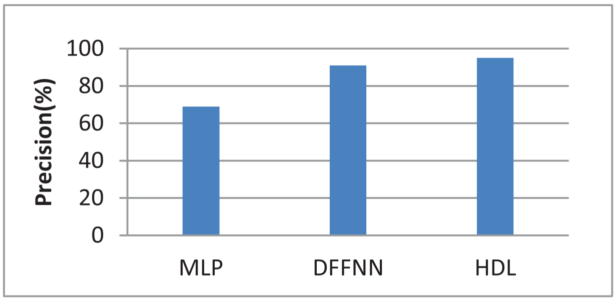

FIGURE 3. Precision comparison results between the existing and proposed student prediction model.

the offered datasets. These findings align with the preceding mistake rate and could be connected to the rule sets generated by the suggested HDL classification model. The results show that, in terms of the f-measure, the proposed HDL strategy performs better than the already-used classification techniques.

The two types of accuracy—sensitivity and specificity—are typically used to judge a model's viability. Fig. 6 displays the proposed and present student prediction models' accuracy values. According to the simulation findings, the proposed HDL model has an accuracy rate of 95.67%, higher than that of the current DFFNN model (93.9%) and the MLP model (71.41%). Statistics demonstrate that the proposed FDL methodology produces better accuracy than current classification methods.

Calculation of Information Criteria: Calculate the AIC, SBIC, and HQIC for your models. The formulas are as follows:

- **AIC (Akaike Information Criterion):**

AIC=2k−2ln⁡(L)\text{AIC} = 2k - 2\ln(L)AIC=2k−2ln(L)

Where kkk is the number of parameters in the model and LLL is the likelihood of the model.

- **SBIC (Schwarz Bayesian Information Criterion):**

SBIC=kln⁡(n)−2ln⁡(L)\text{SBIC} = k \ln(n) - 2\ln(L)SBIC=kln(n)−2ln(L)

Where nnn is the number of observations.

- **HQIC (Hannan-Quinn Information Criterion):**

HQIC=2kln⁡(ln⁡(n)) −2ln⁡(L)\text{HQIC} = 2k \ln(\ln(n)) - 2\ln(L)HQIC=2kln(ln(n))−2ln(L)

Once the optimal hyperparameters were determined through cross-validation, the final HDL model was trained on the entire training set. The final model's performance was evaluated using the ANOVA test set, which was not seen during training or hyperparameter tuning. Table 1 shows an unbiased evaluation of the model's performance.

Cross-validation played a crucial role in validating the effectiveness of the HDL approach. It helped assess the model's robustness, generalization ability, and reliability in accurately predicting student performance. The averaged metrics from K-fold cross-validation provided a comprehensive understanding of how well the HDL approach could identify high-risk students, ensuring the credibility of the

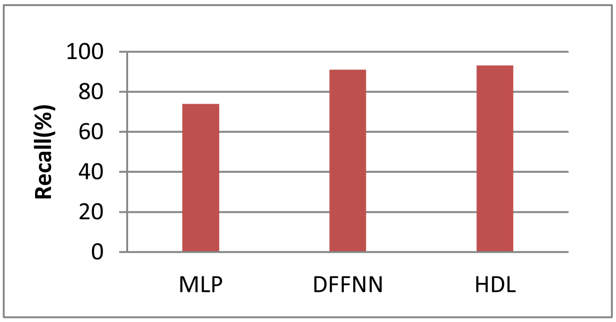

FIGURE 4. Recall comparison results between the existing and proposed student prediction model.

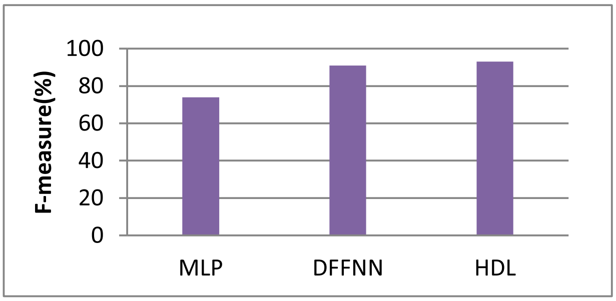

FIGURE 5. F-measure comparison results between the existing and proposed student prediction model.

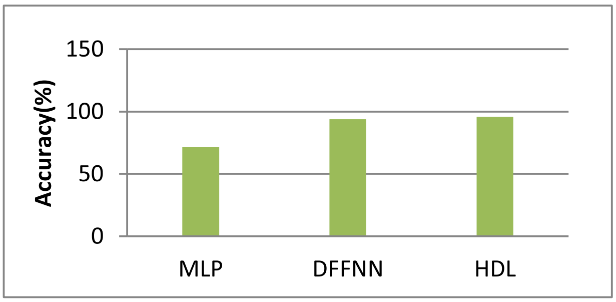

FIGURE 6. Accuracy comparison results between the existing and proposed student prediction model.

TABLE 2. Performance analysis, including ANOVA.

Model/ Statistic	Accuracy	Precision	Recall	F1-Score	p-value (ANOVA)
HDL Model	95.67%	94.24%	96.45%	93.78%	0.28725
MLP Model	71.41%	75.65%	78.92%	76.85%	0.00012
DFFNN Model	93.9%	91.23%	89.75%	91.62%	0.00104

results. In addition to the standard performance metrics of Accuracy, Precision, Recall, and F1-Score, we have included the Akaike Information Criterion (AIC), Schwarz Bayesian Information Criterion (SBIC), and Hannan-Quinn Information Criterion (HQIC) to evaluate the models. Including this information, criteria provide a more robust and scientific basis for model selection, accounting for model complexity and goodness of fit. The results in Table 2 show that Model 1, despite having a slightly lower accuracy, is preferred based on AIC, SBIC, and HQIC values, indicating it achieves a better balance between complexity and fit. While our proposed Hybrid Deep Learning (HDL) approach using ECNN and ResNet models has shown significant accuracy in predicting students' performance, recent studies suggest that Generative Pre-Trained Transformers (GPT) might offer even better performance for time-dependent data. GPT models, which utilize an encoder-decoder architecture, can capture complex patterns and dynamics in the data more effectively. Incorporating GPT models in future studies could potentially enhance prediction accuracy and provide deeper insights into students' learning behaviors

V. CONCLUSION

Academic achievement of students at any professional institution has emerged as management's main concern. Early identification of pupils at risk for underperformance enables management to move quickly to boost those students' performance through additional coaching and counseling. To find the best-performing predictive model, this study proposes an HDL framework to forecast students' performance utilizing ECNN and Resnet model-based classification algorithms. Min-max normalization is used to perform the preprocessing at first. Also, this work employed Butterfly optimization-based feature selection approaches to choose the top features from the dataset connected to students' performance. Eventually, the HDL is created to effectively forecast high-risk pupils in a model based on a VLE. The OULAD is used to assess the proposed model, and the effectiveness of various classifiers is included in this study for comparative evaluations using the metrics of precision, recall, and F-score values. From the experimental results, the proposed HDL model has an accuracy rate of 95.67%, which is higher when compared with the other existing works like the DFFNN model (93.9%) and the MLP model (71.41%). Statistics demonstrate that the proposed FDL methodology produces better accuracy than current classification methods.

A. LIMITATIONS

The swarm-intelligence-based optimization algorithms have yet to be considered for tuning the weight and bias of the parameters of the proposed classifier. This work must discuss the temporal characteristics for forecasting students' should bens and grade models. One limitation of our study is the exclusion of recent advancements in Generative Pre-Trained Transformers (GPT) for predicting time-dependent data. While our HDL approach has demonstrated high accuracy, GPT models might offer superior performance.

B. FUTURE DIRECTIONS

The swarm-intelligence-based optimization algorithms will be introduced to tune the weight and bias of parameters of a proposed classifier to improve classification accuracy further. Furthermore, using temporal characteristics for forecasting students' evaluations and grade models is an area of future research. Time series analysis will be performed using temporal features, and more sophisticated machine learning will be used. Future research should explore integrating GPT models to predict students' performance. By leveraging the encoder-decoder architecture of GPT models, capturing more complex patterns in the data might be possible, thereby improving prediction accuracy.

REFERENCES

[1] E. Ahn, Y. Gil, and E. Putnam-Hornstein, "Predicting youth at high risk of aging out of foster care using machine learning methods," *Child Abuse Neglect*, vol. 117, Jul. 2021, Art. no. 105059, doi: 10.1016/j.chiabu.2021.105059.

[2] R. Dembo and W. Walters, "Innovative approaches to identifying and responding to the needs of high risk youth," *Substance Use Misuse*, vol. 38, nos. 11–13, pp. 1713–1738, Jan. 2003, doi: 10.1081/ja-120024238.

[3] H. H. Severson, H. M. Walker, J. Hope-Doolittle, T. R. Kratochwill, and F. M. Gresham, "Proactive, early screening to detect behaviorally at-risk students: Issues, approaches, emerging innovations, and professional practices," *J. School Psychol.*, vol. 45, no. 2, pp. 193–223, Apr. 2007, doi: 10.1016/j.jsp.2006.11.003.

[4] M. Yağcı, "Educational data mining: Prediction of students' academic performance using machine learning algorithms," *Smart Learn. Environ.*, vol. 9, no. 1, p. 11, Dec. 2022, doi: 10.1186/s40561-022-00192-z.

[5] T. O. Olaleye and O. R. Vincent, "A predictive model for students' performance and risk level indicators using machine learning," in *Proc. Int. Conf. Math., Comput. Eng. Comput. Sci. (ICMCECS)*, Mar. 2020, pp. 1–7, doi: 10.1109/ICMCECS47690.2020.240897.

[6] A. Marwaha and A. Singla, "A study of factors to predict at-risk students based on machine learning techniques," in *Advances in Intelligent Systems and Computing*. Singapore: Springer, 2020, pp. 133–141.

[7] O. Embarak, "Apply machine learning algorithms to predict at-risk students to admission period," in *Proc. 7th Int. Conf. Inf. Technol. Trends (ITT)*, Nov. 2020, pp. 190–195, doi: 10.1109/ITT51279.2020.9320878.

[8] W. Xiao, P. Ji, and J. Hu, "A survey on educational data mining methods used for predicting students' performance," *Eng. Rep.*, vol. 4, no. 5, May 2022, doi: 10.1002/eng2.12482.

[9] S. D. A. Bujang, A. Selamat, and O. Krejcar, "A predictive analytics model for students grade prediction by supervised machine learning," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1051, no. 1, Feb. 2021, Art. no. 012005, doi: 10.1088/1757-899X/1051/1/012005.

[10] L. Ismail, H. Materwala, and A. Hennebelle, "Comparative analysis of machine learning models for students' performance prediction," in *Advances in Digital Science*. Cham, Switzerland: Springer, 2021, pp. 149–160.

[11] S. Gaftandzhieva, A. Talukder, N. Gohain, S. Hussain, P. Theodorou, Y. K. Salal, and R. Doneva, "Exploring online activities to predict the final grade of student," *Mathematics*, vol. 10, no. 20, p. 3758, Oct. 2022, doi: 10.3390/math10203758.

[12] R. Ayienda, R. Rimiru, and W. Cheruiyot, "Predicting students academic performance using a hybrid of machine learning algorithms," in *Proc. IEEE AFRICON*, Sep. 2021, pp. 1–6, doi: 10.1109/AFRICON51333.2021.9571012.

[13] B. S. Kalyani, D. Harisha, V. RamyaKrishna, and S. Manne, "Evaluation of students performance using neural networks," in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, 2020, pp. 499–505.

[14] N. Naicker, T. Adeliyi, and J. Wing, "Linear support vector machines for prediction of student performance in school-based education," *Math. Problems Eng.*, vol. 2020, pp. 1–7, Oct. 2020, doi: 10.1155/2020/4761468.

[15] X. Wang, X. Yu, L. Guo, F. Liu, and L. Xu, "Student performance prediction with short-term sequential campus behaviors," *Information*, vol. 11, no. 4, p. 201, Apr. 2020, doi: 10.3390/info11040201.

[16] K. Neha, J. Sidiq, and M. Zaman, "Deep neural network model for identification of predictive variables and evaluation of student's academic performance," *Revue d'Intelligence Artificielle*, vol. 35, no. 5, pp. 409–415, Oct. 2021, doi: 10.18280/ria.350507.

[17] S. Begum and S. S. Padmannavar, "Genetically optimized ensemble classifiers for multiclass student performance prediction," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 2, pp. 316–328, Apr. 2022, doi: 10.22266/ijies2022.0430.29.

[18] Y. A. Alsariera, Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, and N. Ali, "Assessment and evaluation of different machine learning algorithms for predicting student performance," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, May 2022, doi: 10.1155/2022/4151487.

[19] S. Damuluri, K. Islam, P. Ahmadi, and N. S. Qureshi, "Analyzing navigational data and predicting student grades using support vector machine," *Emerg. Sci. J.*, vol. 4, no. 4, pp. 243–252, Aug. 2020, doi: 10.28991/esj-2020-01227.

[20] A. B. Raut and A. A. Nichat, "Students performance prediction using decision tree technique," *Int. J. Comput. Intell. Res.*, vol. 13, no. 7, pp. 1735–1741, 2017.

[21] E. A. Amrieh, T. Hamtini, and I. Aljarah, "Mining educational data to predict student's academic performance using ensemble methods," *Int. J. Database Theory Appl.*, vol. 9, no. 8, pp. 119–136, Aug. 2016, doi: 10.14257/ijdta.2016.9.8.13.

[22] A. Siddique, A. Jan, F. Majeed, A. I. Qahmash, N. N. Quadri, and M. O. A. Wahab, "Predicting academic performance using an efficient model based on fusion of classifiers," *Appl. Sci.*, vol. 11, no. 24, p. 11845, Dec. 2021, doi: 10.3390/app112411845.

[23] M. Adnan, A. Habib, J. Ashraf, S. Mussadiq, A. A. Raza, M. Abid, M. Bashir, and S. U. Khan, "Predicting at-risk students at different percentages of course length for early intervention using machine learning models," *IEEE Access*, vol. 9, pp. 7519–7539, 2021, doi: 10.1109/ACCESS.2021.3049446.

[24] S. G. K. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *Int. Adv. Res. J. Sci. Eng. Technol.*, vol. 2, no. 3, pp. 20–22, Mar. 2015, doi: 10.17148/IARJSET.2015.2305.

[25] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019, doi: 10.1007/s00500-018-3102-4.

[26] S. Targ, D. Almeida, and K. Lyman, "ResNet in ResNet: Generalizing residual architectures," in *Proc. Int. Conf. Learn. Represent Workshop*, San Juan, Puerto Rico, 2016, pp. 1–7.

[27] J. Wu, *Introduction to Convolutional Neural Networks*. Nanjing, China: Nanjing Univ., 2017.

JAFAR ALI IBRAHIM SYED MASOOD received the Bachelor of Technology degree in information technology from the Syed Ammal Engineering College, Ramanathapuram, the Master of Technology degree in computer systems and networks from the Dr. M. G. R. Educational and Research Institute, Chennai, Tamil Nadu, India, and the Doctor of Philosophy degree in information and communication engineering from Anna University, Chennai.

He was affiliated with Anna University. He was a full-time Research Fellow in information and communication engineering with Anna University. Currently, he works as a Faculty Member with the Internet of Things Department, School of Computer Science and Engineering, Vellore Institute of Technology (Deemed to be University), Vellore, Tamil Nadu. He possesses nearly seven years of industrial experience, both administrative and technical perception in IT security, bio-metric deployment, IT administration, network security, web services and security, and service-oriented architecture environments. Also, he possesses six years of research and teaching experience in health applications, genetic data analysis, computational and machine intelligence, and IoT-enabled recommender systems. He has presented his paper at many international conferences and published 86 journal articles. He published more than 100 national and international patents and holds 38 Patent grants. His research interests include cloud computing, clinical research, medical informatics, the Internet of Things, intelligent systems development, machine learning, ontology development, big data, and data mining.

Dr. Masood serves as a reviewer, an editorial board member, a technical committee member, and an advisor in many international conferences and journals. He is a consultant in some IT industries from Malaysia and U.K. He is an Active Member of the Academic Council, Board of Studies, Board of Research, and Governing Council, for various reputed institutions in India and Abroad. Also, he is a reviewer of some reputed international journals and ACM conferences. He is a lifetime member of various reputable and autonomous technical bodies from different countries. He visited Japan, Hong Kong, Malaysia, Singapore, Thailand, Indonesia, South Korea, U.K., Philippines, Sri Lanka, and Gulf countries for research.

N. S. KALYAN CHAKRAVARTHY received the Bachelor of Engineering degree in computer science and engineering, the Master of Technology degree in computer science and engineering, the master's degree in artificial intelligence for leaders, and the Doctor of Philosophy degree in computer science and engineering.

He is currently the Director for data science, business systems, AI, and ML of the QIS College of Engineering and Technology, Ongole, Andhra Pradesh, India. He has three decades of experience in both industry and academia. As a Technocrat, he served various domain-centric industries in various positions and capacities. He has vast functional and technical expertise in the product development life cycle. With a passion for serving the community, he has presented more than 25 papers at reputed international conferences and published twenty articles in reputed journals. He also has 52 patents.

DAVID ASIRVATHAM received the B.Sc. Ed. (Hons.) and Post-Graduate Diploma degrees in computer science from the University of Malaya, the M.Sc. degree in digital systems from Brunel University, U.K., and the Ph.D. degree from Multimedia University. He has been lecturing and managing ICT projects for the past 30 years. He is currently the Executive Dean of the Faculty of Innovation and Technology, Taylor's University, Malaysia. His area of expertise will include digital neural networks and e-learning technologies, and recently he has done some work on big data analytics and AI.

MOHSEN MARJANI received the master's degree in information technology (multimedia computing) from Multimedia University (MMU), Malaysia, and the Ph.D. degree in computer science from the University of Malaya (UM). Since 1999, he has taught mathematics and IT-based subjects in public and private institutions, including universities, pre-universities, high schools, and primary schools. He has experience collaborating with and working for several IT companies as an IT Consultant, a Senior Web and a Mobile Developer, the Tech Lead, the Project Manager, and the CTO. He currently works as a Senior Lecturer with the School of Computer Science, Taylor's University, Malaysia. He has published multiple research articles in refereed international journals. His research interests include big data, data analytics, AI, machine learning, deep learning, image processing, and the IoT.

DALIA ABDULKAREEM SHAFIQ was born in Baghdad, Iraq, in May 1995. She received the B.Sc. and M.Sc. degrees in computer science from Taylor's University, Malaysia, in 2019 and 2021, respectively, where she is currently pursuing the Doctor of Philosophy (Ph.D.) degree with the School of Computer Science. She tutors undergraduate students with the School of Computer Science, Taylor's University. Her research interests include recommendation systems, machine learning, educational data mining, big data analytics, and data science. Other research interests include distributed computing, cloud computing, and simulations.

SRINU NIDAMANURI received the B.Tech. degree in computer science engineering from JNTUK, in 2007, and the M.Tech. degree in computer science and engineering from the QIS College of Engineering and Technology, Ongole, in 2011. He is currently pursuing the Ph.D. degree with the Bharath Institute of Higher Education and Research. He has 11 years of experience as an Assistant Professor. He is an Associate Professor with the CSE Department, QIS College of Engineering and Technology. His research interests include data science, machine learning, deep learning, and image processing.