International Journal of Innovative Research and Scientific Studies, 8(6) 2025, pages: 1612-1626

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Gamification elements-based engineering students' learning behavior analysis and performance modelling system using ZSH-FUZZY and Deep-JRGNN

Swati Joshi[1*], Sanjay Kumar Sharma[2]

[1,2]*Department of Computer Science, Banasthali Vidyapith, Rajasthan, India.*

Corresponding author: Swati Joshi (*Email: cse.swatijoshi@gmail.com*)

Abstract

Educators can identify the students who are at risk of underperformance by predicting their performance. However, the existing works didn't concentrate on specific gamification elements for accurate performance modelling. Therefore, this paper presents gamification elements-based engineering students learning behavior analysis and performance modelling using Deep-JRGNN and ZSH-Fuzzy. Initially, the student's performance data are taken; then, they are pre-processed. Afterward, binning and hexbin plot construction are performed. Then, the features are extracted from the hexbin plot. Likewise, pre-processed data are clustered by employing WCTQEK-Means. For the clustered data, the temporal behavioral analysis is done by utilizing the DACWT. Next, the temporal features and behavioral features are extracted. Afterward, the correlation analysis is performed. Then, engineering students' learning behavior is analyzed by using the percentile rank calculation and ZSH-Fuzzy. Likewise, the gamification elements are considered from the questionnaires. For that, percentile calculation is done and classes are identified by using ZSH-Fuzzy. Then, labelling is done by using ZSH-Fuzzy. Finally, the performance modelling is performed by using Deep-JRGNN, in which the LIME-based DeepXplainer provides a deep explanation of the outcomes. The results proved that the proposed model achieved a high accuracy of 98.74%, which outperformed conventional methods.

Keywords: Deep Jacobian-Pathwise recurrent growing-cosid neural network (Deep-JRGNN), Discrete additive Cholesky wavelet transform (DACWT), Gamification elements, Local interpretable model-agnostic explanation (LIME), Missing value imputation (MVI) and learning behavior, Wasserstein ChordTsallis quantum entropy k-means (WCTQEK-Means), ZS-shaped hyperbolic Fuzzy (ZSH-Fuzzy).

DOI: 10.53894/ijirss.v8i6.9995
Funding: This study received no specific financial support.
History: Received: 25 June 2025 / **Revised:** 29 July 2025 / **Accepted:** 31 July 2025 / **Published:** 19 September 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

Generally, the analysis of engineering student's learning behavior involves how engineering students engage with learning activities [1]. For every institute, the most significant need is the performance of their student [2]. Educational institutions store a massive amount of information for tracking students, faculty, and courses. This information helps in predicting the student's performance [3, 4]. Likewise, performance prediction of engineering studentshelps to identify which teaching practices have a positive impact on students' learning [5]. Recently, many Artificial Intelligence (AI) techniques have been employed to predict engineering student's performance.

Machine Learning (ML) algorithms, such as RF and logistic regression were utilized to predict the performance of engineering students [6, 7]. Also, some prevailing works employed Bayesian optimization and ANN for predicting the student's academic performance [8, 9]. Likewise, the genetic programming technique is used by some conventional works for analyzing the student's behavior [10]. However, the prevailing works didn't concentrate on specific gamification elements for accurate performance modelling. Therefore, gamification elements-based engineering students' learning behavior analysis and performance modelling system by using Deep-JRGNN and ZSH-Fuzzy is proposed.

1.1. Problem Statement

- None of the existing works concentrated on specific gamification elements for accurate performance modelling.
- In prevailing [11] the suggestions were inadequate to address the effective performance prediction of diverse engineering student groups.
- The potential relationship between the variables was not identified in existing [12].
- The conventional [13] had poor performance predictive tools; also, it had a black-box nature.
- Most of the existing works considered only subject mark features for analyzing the engineering students' learning behavior.

1.2. Objective

- Gamification elements are gathered from questionnaires and analyzed using percentile rank calculations. Based on these ranks, classification is performed using ZSH-Fuzzy, which helps in categorizing elements effectively. This approach enhances the accuracy of classification by leveraging fuzzy logic principles within the ZSH framework.
- Building on the classification of gamification elements using ZSH-Fuzzy, WCTQEK-Means is introduced to cluster student groups based on these classified elements. This clustering method ensures that students with similar engagement patterns and preferences are grouped together, optimizing the effectiveness of gamified learning strategies.
- Kendall's Tau is employed to analyze the correlation between these classified elements and clustered student groups. This correlation analysis helps in understanding the relationships between different gamification elements and student engagement patterns, ensuring a data-driven approach to refining gamified learning strategies.
- Deep-JRGNN and LIME are introduced for accurate performance prediction of engineering students. Deep-JRGNN leverages deep learning and graph neural networks to model complex relationships, while LIME provides explainability by identifying key factors influencing student performance. This predictive framework ensures a comprehensive understanding of how gamification elements impact student learning outcomes.
- DACWT is utilized to perform temporal behavioral analysis. DACWT helps track changes in student engagement and learning behavior over time, providing insights into how gamification elements influence long-term academic performance. This analysis enhances the adaptability of gamified learning strategies by identifying evolving patterns and trends in student behavior.

This paper is organized as: Section 2 describes the related works, Section 3 illustrates the proposed methodology, Section 4 conveys the result and discussion, and Section 5 concludes the proposed model with future scope.

2. Literature Survey

Li and Liu [11] presented a framework named performance prediction for higher education students. Here, the Deep Neural Network (DNN) was used to predict the performance of higher education students. Thus, the model effectively reduced the difficulties in education. Yet, theprovided suggestions were inadequate for diverse engineering student groups' performance prediction.

Ayaz, et al. [14] presented a report which is the second of a two-part series that examines a wide range of novel methodologies for investigating brain health and function. While the last report concentrated on neurophotonic tools that were primarily applicable to animal studies, this report focuses on optical spectroscopy and imaging methods that are useful to noninvasive human brain studies.Researcher outlined current cutting-edge technology and software breakthroughs, investigate their most recent influence on neuroscience and clinical applications, propose possibilities for innovation, and forecast future directions.

Kanetaki, et al. [12] explored a performance prediction model for sustainable engineering education. Here, the research employed the generalized linear autoregressive model for predicting the grades of engineering students. Also, the model was well suited for all aspects of the academic process. Furthermore, this work failed to identify the potential relationship between the variables.

Luo, et al. [15] analyzes student data from Sina Weibo to develop a multimodal fusion model for detecting early student depression. It compares early and late fusion methods to typical Natural Language Processing models and improves accuracy by 3% over 100 cycles. The study found that standardizing merely structured data without neural network mapping affects prediction effectiveness. This study evaluates the capacity of multimodal fusion models to efficiently detect early indicators of student depression and establishes the foundation for future research on model interpretability for early student depression detection.

Priyambada, et al. [13] described a model for understanding the student's learning behavior. Here, the profile-based cluster evolution analysis was utilized to analyze the learning behavior of students. Also, the presented model could be employed as a recommendation to academic stakeholders. Yet, the research had poor performance predictive tools; also, it had a black-box nature.

Leelaluk, et al. [16] suggested a learning activities-based engineering student's performance prediction model. Here, the Attention-based Artificial Neural Network (Attn-ANN) was employed for predicting engineering student's performance. Thus, the research effectively classified the at-risk students. However, the model had insufficient data and limited features.

Giannakas, et al. [17] propounded an early prediction of a team-based academic performancemodel. Here, the DNN algorithm was employed to predict the academic performance in software engineering. Hence, the research obtained high accuracy, and also it assisted individuals in developing their skills. But this work took more time for training and had overfitting issues.

Massa, et al. [18] conducted a pilot test to analyze the effectiveness of the PHOTON PBL Challenges and associated assessment methodologies in photonics technology education. The study surveyed 21 photonics technology students and four teachers from four community institutions. In the study assessed the impact of PHOTON PBL Challenges on students' problem-solving skills, motivation, self-efficacy, and metacognitive capacity. They also completed a pre- and post-online survey (MSLQ) at the start and end of each semester. Student work samples, including concept maps, whiteboards.

Firoz, et al. [19] studied the impact of mental stress on students' academic performance, social interactions, and emotional well-being. In this study, they used train and test cases to detect mental stress in a sample size of 2116 students. They used six classification algorithms (Decision Tree Classifier, Random, Forest Classifier, SVC, KNNClassifier, Multinomial NB, and K-Nearest Neighbors Regressor) on the dataset. The Decision Tree Classifier and Random Forest Classifier achieved the highest test result of 0.99, outperforming the other six approaches. It is found that algorithms such as Decision Tree Classifier and Random Forest Classifier can improve the mental health of university students.

Smith, et al. [20] proposed a model that eliminates the need for post-training fixed point optimization, allowing us to investigate the SLDS's (Switching linear dynamical system) learned dynamics at any point in state space. It also extends SLDS models to continuous manifolds of switching points, sharing properties across switches. They test the model's utility using two simulated tasks related to past work reverse engineering RNNs. They next demonstrate that their model can be utilized as a drop-in in more complicated designs, such as LFADS, and use this LFADS hybrid to assess single-trial spiking activity from a nonhuman primate's motor system.

Fodor and Kamath [21] tested different 2-D denoising algorithms on test photos that have been contaminated by additive Gaussian noise.They investigated global, level-dependent, and subband-dependent implementations of these approaches. Their results show that the SureShrink and BayesShrink approaches regularly beat the other wavelet-based strategies in terms of denoising quality, as measured by mean squared error. In contrast, they discovered that a combination of simple spatial filters resulted in grainier images with smoother edges, while the inaccuracy was less than in wavelet-based approaches.

Wang, et al. [22] discussed that Discrete Wavelength Transform (DWT) plays a significant role in digital signal processing. Daubechies order 4 wavelet transform (DB4) is chosen to discuss in this work. This study analyzes the benefits of DWT and implements a three-level Mallat algorithm with selected db4 low-pass and high-pass filters. This work introduces and implements an orthogonal two-channel db4 filter with a lattice structure. It also presents an enhanced Mallat structure. This paper uses EEG data to verify the intended DWT. The design is implemented in Verilog, simulated using Modelsim, and verified on the ISEI3.3 platform.

Lubis, et al. [23] research indicates that the Entropy approach can determine the initial centroid value in K Means. The Entropy method's centroid value provides effective data analysis during training. The entropy approach generates centroid values that can represents rice productivity with only 12 training data points and 2 iterations. However, the learning process is becoming more effective. The initial centroid value is more effective than normal K Means, which uses a random number.

Kumarakulasinghe, et al. [24] study compared LIME(Local Interpretable Model-agnostic Explanation) to physician responses to understand how a "black-box" machine learning algorithm predicted sepsis or not. LIME explanations were well-received by physicians, indicating positive results. Additionally, physicians demonstrated cautiously optimistic trust and reliance on LIME. To promote widespread implementation of "black-box" machine learning predictive techniques in healthcare, more research is needed to improve LIME and identify aspects that increase physician trust.

Kurniadi, et al. [25] described a prediction technique based on K-nearest neighbor and a strength and problem questionnaire. They employed the K-nearest neighbor algorithm to identify troublesome pupils from a dataset of 100 persons. Predictive computation consists of four phases: simulation, accuracy assessment, data cleansing, and collecting. Using the rapid application development technique, the system is being built with the student's condition mapped using the Strengths and Difficulties Questionnaire. These initiatives aim to support classroom instructors by teaching children about different learning styles, assisting parents in better understanding their child's personality, and assisting counseling teachers in implementing an early warning system.

Marjan, et al. [26] devised an educational data mining-based approach for evaluating and improving university students' programming skills. The suggested educational data mining method included two necessary modules for classification and the learning process. The categorization module forecasts a student's present condition, while the learning module generates relevant suggestions and feedback to improve learning quality. The study used six popular machine learning methods for classification: Random Forest, SVM, k-NN, Naïve Bayes, Decision Tree, and ANN. The researcher created a genuine dataset and then evaluated each algorithm's performance on the real dataset using goodness of fit and assessment criteria.

Bujang, et al. [27] provide a detailed study of machine learning algorithms for estimating students' final marks in first-semester courses. Their efforts were primarily focused on two components. Using a dataset of 1282 students' course results, researchers evaluated the performance of six well-known machine learning approaches: decision trees, random forests, k-nearest neighbor, support vector machines, naïve bayes, and logistic regression. The following phase required developing a prediction model for many classes. To prevent misclassification and overfitting caused by imbalanced multi-classification, the model used the Synthetic Minority Oversampling Technique.

Chen, et al. [28] used big data mining and qualitative analysis techniques to extract insights about students' learning experiences from unstructured social media platforms like Twitter and Facebook. The author analyzed twitter data from engineering students to better understand their learning experiences. This paradigm enables administrators, practitioners, and decision makers to leverage student study experience data to improve teaching and learning processes.

3. Proposed Engineering Student's Performance Modelling Framework

Here, the Deep-JRGNN is introduced to predict the engineering student's performance. The diagrammatic representation of the proposed model is shown in Figure 1.

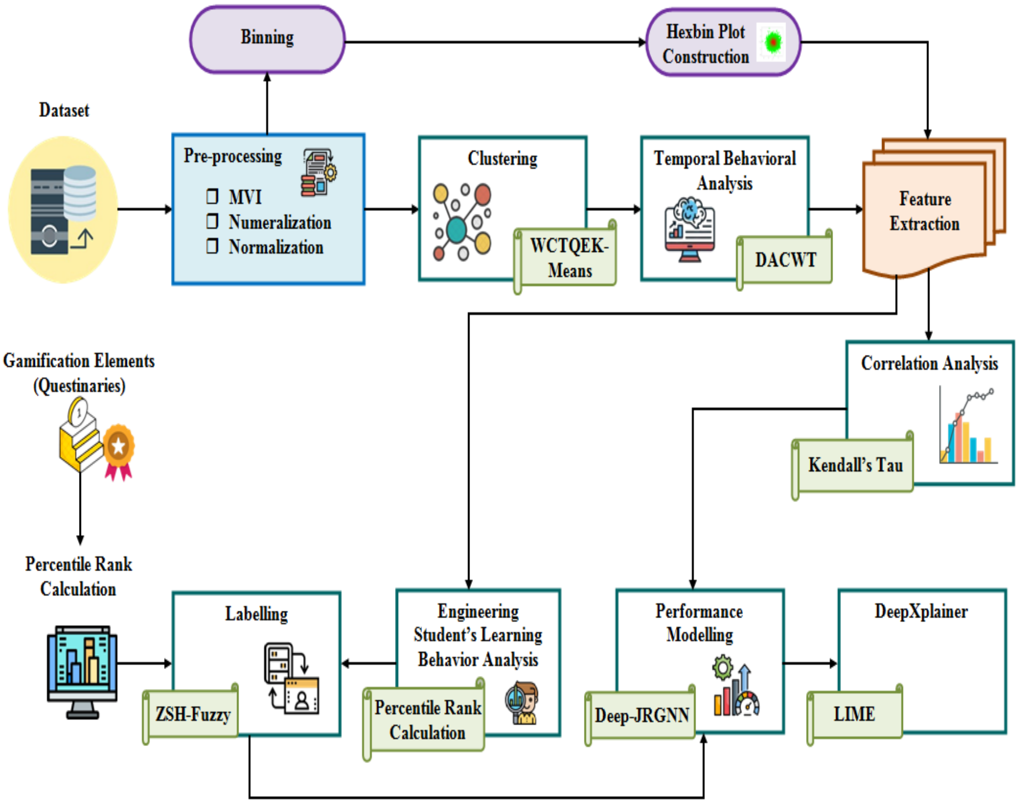


Figure 1.
Diagrammatic representation of the proposed model.

3.1. Dataset

Primarily, the "Student Performance Prediction Dataset" is taken to train the proposed model. The student performance data(Q_{κ}^{data}) is defined as,

$$Q_{\kappa}^{data} = \{Q_1^{data}, Q_2^{data}, Q_3^{data}, \dots\dots, Q_t^{data}\} \quad where \quad \kappa = (1\ to\ t) \qquad (1)$$

Where, Q_t^{data} indicates the number of (Q_{κ}^{data}).

3.2. Pre-Processing

Then, the(Q_{κ}^{data})arepreprocessed. Firstly, the missing values in the (Q_{κ}^{data})areimputed by using the mean formula,

$$M_{im} = \frac{\sum Q_{\kappa}^{data}}{t} \qquad (2)$$

Here, M_{im} specifies the missing value imputed data. Then, the M_{im} are numeralized. Here, the string values are converted into numerical values and are indicated as λ. Afterward, the λ are converted into 0 to 1 values in the normalization process. Here, the Z-score is employed for normalization. Thus, the normalized data (N) is given as,

$$N = \frac{\lambda - \mu}{\sigma} \tag{3}$$

Where, μ depicts the mean value, and σ denotes the standard deviation value. The pre-processed data is represented as $\wp_s$.

3.3. Binning and Hexbin Plot Construction

Subsequently, binning is done in which the $\wp_s$ is converted into small bins and is denoted as b_ε. From b_ε, the hexbin plot is constructed to distinguish the diverse student behaviors. The constructed hexbin plot is defined as He_{plot}.

3.4. Clustering

Likewise, by using WCTQEK-Means, the $\wp_s$ are clustered based on the similar behavioral patterns of students. K-Means proficiently handles large datasets with high dimensionality. Nevertheless, K-Means struggled to cluster the varying density clusters. Also,

$$Tq = -f(\varphi \;\; log \;\; \varphi) * Bo \cdot \frac{1-\sum \wp_s}{r-1} \tag{4}$$

Where, f indicates the trace of a matrix, φ is the density matrix, Bo specifies the Boltzmann constant, and r indicates the entropic index.

$$C_u \xrightarrow{Tq} (C_1 + C_2 + C_3 + \cdots \ldots + C_v) \tag{5}$$

Here, C_v signifies the number of initialized centroids (C_u). Next, Wasserstein Chord distance (D) is calculated as,

$$D(C_u, \wp_s) = \sum_{u=1,s=1}^{v,e} |C_u - \wp_s| \bullet \sqrt{2(1 - cos(\theta))} \tag{6}$$

Where, θ is the angle between the vectors. Then, for all data, the average is computed. Next, the new centroid is estimated for all clusters.

$$Nc = \frac{\sum_{u=1,s=1}^{v,e} A_{us} \wp_s}{\sum_{u=1,s=1}^{v,e} \wp_s} \tag{7}$$

Where, Nc signifies the new centroid, and A_{us} is the average of data. Afterward, each node is reassigned to the new closest centroid of each cluster. The aforementioned steps are continued until the optimal clusters. The clustered student data are represented as $\Im_a$.

Pseudocode for WCTQEK-Means

Input: Preprocessed data $(\wp_s)$

Output: Clustered student data $(\Im_a)$

Begin

    Initialize $(\wp_s)$

    For each $(\wp_s)$

        Compute $Tq = -f(\varphi \;\; log \;\; \varphi) * Bo \cdot \frac{1-\sum \wp_s}{r-1}$

        Initialize centroids (C_u) based on Tq

        Find $D(C_u, \wp_s) = \sum_{u=1,s=1}^{v,e} |C_u - \wp_s| \bullet \sqrt{2(1 - cos(\theta))}$

        Estimate average for all data

        Evaluate $Nc = \frac{\sum_{u=1,s=1}^{v,e} A_{us} \wp_s}{\sum_{u=1,s=1}^{v,e} \wp_s}$

        Reassign each node

        End For

    Obtain $(\Im_a)$

End

Thereafter, the temporal behavioral analysis is done for the $(\Im_a)$.

3.5. Temporal Behavioral Analysis

Then, based on $(\Im_a)$, the temporal behavioral analysis is performed by using DACWT. Discrete Wavelet Transform (DWT) is well-suitable for analyzing temporal behavioral data owing to its adaptability. But, DWT selects significant coefficients for analysis, thus resulting in information loss. Therefore, the Additive Cholesky Decomposition is employed in DWT.

DWT (AB) of $(\Im_a)$ is identified by passing the $(\Im_a)$ through the series of filters. Firstly, $(\Im_a)$ are passed through the low pass filter.

$$AB(h) = (\Im_a * j)[h] \rightarrow \sum_{n=-\infty}^{\infty} \Im_a[n] \cdot j(h-n) \tag{8}$$

Where, h represents the level count, j is the impulse response, and n defines the shift parameter. Then, $(\Im_a)$ are decomposed by employing Additive Cholesky Decomposition.

$$E(\mathfrak{I}_a) = En_{\mathfrak{I}_a} * \Theta\Theta^{\Gamma}(\psi_\tau + \hbar_\tau + \Re_\tau) \quad (9)$$

Where, $En_{\mathfrak{I}_a}$ indicates the entropy function, Θ specifies the lower triangular matrix, Γ defines the transpose term, and ψ_τ, $\hbar_\tau$, and $\Re_\tau$ express the trend, seasonal, and residual components, respectively. Then, outcomes of the low pass filter and decomposition are given to the new low pass filter and decomposition.

$$AB_{low}[h] = \sum_{n=-\infty}^{\infty} \mathfrak{I}_a[n] \cdot j(2h-n) \quad (10)$$

$$AB_{high}[h] = \sum_{n=-\infty}^{\infty} \mathfrak{I}_a[n] \cdot E(2h-n) \quad (11)$$

Thereafter, the sub-sampling operator is estimated and applied in the low-pass filter and decomposition process.

$$(AB \downarrow n)[h] = AB(nh) \quad (12)$$

$$AB_{low} = (\mathfrak{I}_a \times j) \downarrow 2 \quad (13)$$

$$AB_{high} = (\mathfrak{I}_a \times E) \downarrow 2 \quad (14)$$

Lastly, obtained temporal behavioral analysis outcomes are signified as ϑ_z.

3.6. Feature Extraction

Afterward, the features like point density, concentration area, and study time vs stress level are extracted from He_{plot} and are denoted as F_q. Likewise, the temporal and behavioral features like active hours, duration,grade, assignment submissions, and project works are extracted from $\vartheta_z$and are represented as β_c. Thus, the features extracted from both He_{plot} and ϑ_z are considered as W_{ab}.

3.7. Correlation Analysis

Thereafter, based on W_{ab}, the correlation between the variables (i.e., attendance(W_{att}) and grade(W_{gra})) are identified for efficient performance modelling. Kendall's Tau(K) is used for correlation analysis.

$$K = \frac{W_{att}-W_{gra}}{W_{att}+W_{gra}} \quad (15)$$

The obtained correlation values are specified as G_{bc}.

3.8. Engineering Student's Learning Behavior Analysis

Next, engineering students' learning behaviors are analyzed based on the W_{ab}. Firstly, the percentile rank calculation is done for the grade factor (i.e., AA, BA, BB, CB, CC, DC, DD, Fail), which is taken from W_{ab}. The percentile rank formula (P) is determined as,

$$P = \frac{m}{J} \times 100 \quad (16)$$

Where, m indicates the number of values less than the given value, and J implies the total number of values.The calculated percentile rank for the grade factor is denoted as g_p. Then, the classes are identified based on the g_p by employing ZSH-Fuzzy. Fuzzy provides effective solutions to complex issues. Moreover, Fuzzy has tuning difficulty of the membership function. Therefore, the ZS-shaped hyperbolic membership function is utilized in Fuzzy.
Usually, the fuzzy rules ($\aleph$) are constructed by employing If-Then rules.

$$\aleph = \begin{cases} if\ g_p < 25 & low \\ if\ 25 \leq g_p \leq 75 & medium \\ if\ g_p \geq 75 & high \end{cases} \quad (17)$$

Here, low, $medium$, and $high$ are the obtained classes. Then, the ZS-shaped hyperbolic membership function is evaluated to avoid the tuning difficulty of the membership function in Fuzzy.

$$X = \begin{cases} 0 & if\ g_p \leq x \\ 1 - \frac{2\tanh\left(v\left(\frac{g_p-x}{y-x}\right)^2\right)+1}{2} & if\ x \leq g_p \leq \frac{x+y}{2} \\ \tanh\left(v\left(\frac{y-g_p}{y-x}\right)^2\right)+1 & if\ \frac{x+y}{2} < g_p \leq y \\ 1 & if\ g_p > y \end{cases} \quad (18)$$

Where, v, x, and y represent the constant values. The rule generation unit performs interference operations in ZSH-Fuzzy. The($\aleph$) are converted into crisp data (U) in fuzzification(Fu) and is denoted as,

$$Fu = (\aleph \to U) \quad (19)$$

Next, the membership function is plotted regarding the crisp data (U). Similarly, the (U) are converted into ($\aleph$) in defuzzification(Df) and is written as,

$$Df = (U \to \aleph) \quad (20)$$

Eventually, the obtained classes(η) are given as,

$$\eta = (low, high, medium) \quad (21)$$

Then, (η) are subjected for labelling process.

3.9. Gamification Elements

Likewise, the gamification elements like points, badges, leader boards, levelling systems, and rewards are considered from the questionnaires and are denoted as I_{φ}.

3.10. Percentile Rank Calculation

Then, percentile rank calculation and rule generationare performed for I_{φ} by using the steps explained in Section 3.8. Finally, the obtained classes (R) are given as,

$$R = \langle low,\ medium,\ high \rangle \qquad (22)$$

Afterward, (R) are fed for labelling.

3.11. Labelling

Then, labelling is performed based on (η) and (R) by utilizing ZSH-Fuzzy. The process of ZSH-Fuzzy is explained in Section 3.8. The fuzzy rules(Ω) are given as,

$$\Omega = \begin{cases} if\ \eta == low\ \&\&\ R == low & low \\ if\ \eta == medium\ \&\&\ R == medium & medium \\ if\ \eta == high\ \&\&\ R == high & high \\ if\ \eta == low\ \&\&\ R == high & Calculate\ P\ and\ set\ rules \\ if\ \eta == medium\ \&\&\ R == high & Calculate\ P\ and\ set\ rules \\ \vdots & \vdots \\ if\ \eta == high\ \&\&\ R == medium & Calculate\ P\ and\ set\ rule \end{cases} \qquad (23)$$

Thus, the labelling outcomes,(L)such as low, high, and medium are given to Deep-JRGNN to predict the performance of engineering students.

3.12. Performance Modelling

By employing Deep-JRGNN, the performance modelling is done based on the (L)andG_{bc}. Deep Recurrent Neural Network (Deep-RNN) makes accurate predictions by learning from past experiences. But, Deep-RNN had a vanishing gradient problem.Also, training of the Deep-RNN is difficult. Therefore, the growing-cosid activation functionandthe Jacobianpathwise regularization technique are employed in Deep-RNN. The Deep-JRGNN classifier diagram is displayed in Figure 2.

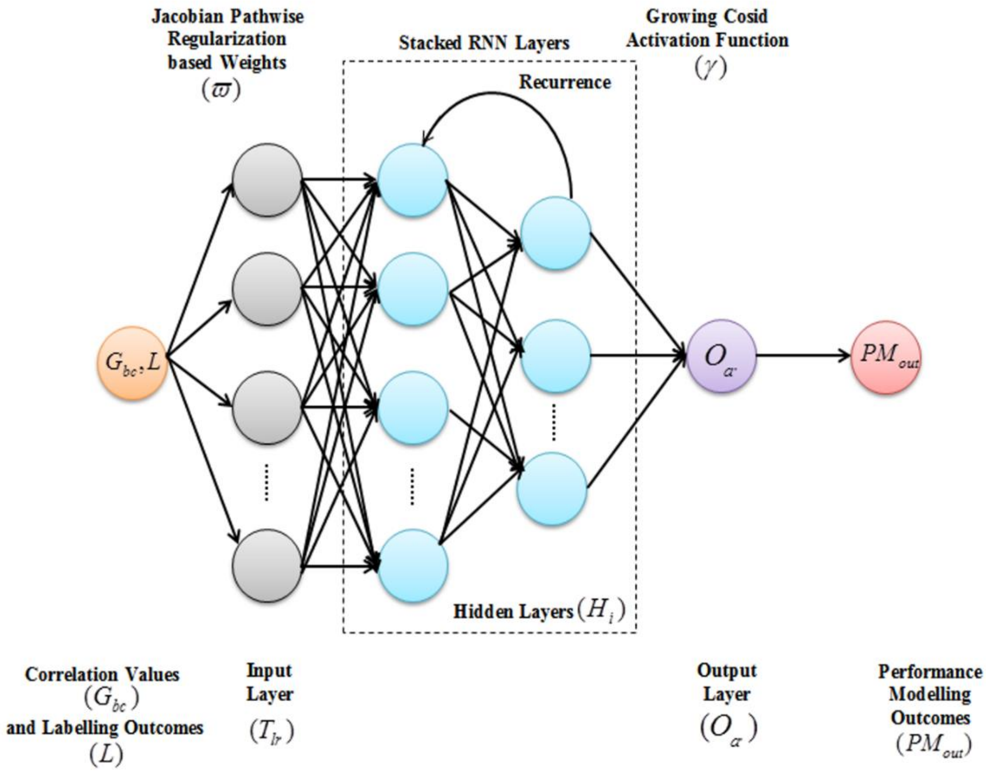


Figure 2.
Deep-JRGNN classifier.

Primarily, the input layer (T_{lr}) gathers (L) and G_{bc} as an input. Then, (T_{lr}) are fed to stacked RNN layers for further process in which the hidden state is updated according to the current input and previous hidden state (H_{i-1}). The hidden state (H_i) is formulated as,

$$H_i = \gamma \cdot (H_{i-1}, T_{lr}) * \varpi + B \quad (24)$$

Where, B specifies the bias, γ implies the growing-cosid activation function, and ϖ denotes the jacobian-pathwise regularization-based weights. Thus, the ϖ is given as,

$$\varpi = \partial^2 \int_0^{Tr} \|Ja(T_{lr}) * \tilde{\varphi}(T_{lr})\|^2 \, dT_{lr} \quad (25)$$

Where, ∂ implies the regularization strength, Tr denotes the training time, $Ja(T_{lr})$ specifies the Jacobian matrix, and $\tilde{\varphi}$ is the derivative of the parameter. Similarly, γ is expressed as,

$$\gamma = (cos(T_{lr}) - T_{lr}) * (sin(T_{lr}) - q'T_{lr}) \quad (26)$$

Where, q' indicates the fixed parameter. Then, the output layer$(O_{a'})$is formulated as,

$$O_{a'} = \varpi * H_i + B \quad (27)$$

Thus, the performance modelling outcomes(PM_{out}) are given as,

$$PM_{out} = [l, w, k] \quad (28)$$

Here, l defines the performance as low, w implies the performance as medium, and k defines the performance as high. Next, the(PM_{out}) are given to DeepXplainer. Here, LIME-based DeepXplainer provides a deep explanation ofthe(PM_{out}). Initially, the instances $(PM_{out})$are selected. Then, they are augmented to create a new dataset of similar instances $(\tilde{G}_{t'})$. Next, the weight value$(\omega\tau)$ is assigned to each instance by using the kernel function.

$$\omega\tau = exp\left(-\frac{\|PM_{out} - \tilde{G}_{t'}\|^2}{\varsigma^2}\right) \quad (29)$$

Here, ς denotes the weight parameter. Afterward, the local model is trained by employing the weighted dataset. Lastly, the obtained description regarding the engineering student's performance is indicated as $Dp_{\hat{\lambda}}$.

Pseudocode for Deep-JRGNN

Input: Labelling outcomes(L) and Correlation values (G_{bc})
Output: Description regarding the engineering student's performance $(Dp_{\hat{\lambda}})$

```
Begin
    Initialize(ϖ),(B)
    For (L) and(G_bc)
        Estimate input layer (T_lr)
        Perform stacked RNN layers
        Computehidden state
            H_i = γ · (H_{i-1}, T_lr) * ϖ + B
        Find
            ϖ = ∂^2 ∫_0^{Tr} ||Ja(T_lr) * φ̃(T_lr)||^2 dT_lr
        Discoverγ = (cos(T_lr) - T_lr) * (sin(T_lr) - q'T_lr)
        EvaluateO_a' = ϖ * H_i + B
        Get PM_out = [l, w, k]
        Select(PM_out)
        Augment to create (G̃_t')
        Assignωτ = exp(-||PM_out - G̃_t'||^2 / ς^2)
        Train local model using weighted dataset
    End For
    Obtain(Dp_λ̂)
End
```

Thus, the Deep-JRGNN effectively predicted the engineering student's performance and provided a description of the performance outcomes.

4. Result and Discussion

In this section, the performance analysis is done to prove the efficacy of the proposed model and is implemented in the working platform of PYTHON.

4.1. Dataset Description

Here, the "Student Performance Prediction Dataset" is used to assess the proposed model. This dataset is collected from publicly available sources and the dataset link is mentioned under the reference section. The dataset consists of 145 student's performance information. From that, 80% of data are employed for training and 20% of data are used for testing.

4.2. Performance Assessment

The performance assessment of the proposed techniques and conventional models are described as follows,

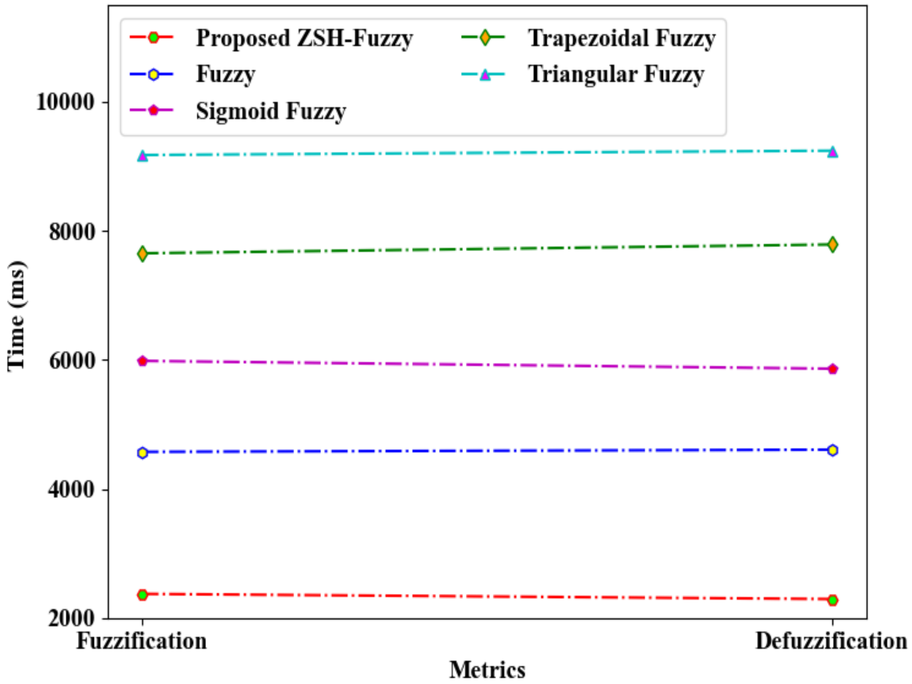


Figure 3.
Graphical analysis of the proposed model and prevailing methods.

Figure 3 displays the graphical analysis of the proposed ZSH-Fuzzy and prevailing methods. Here, the proposed ZSH-Fuzzy took a less fuzzification and defuzzification time of 2378ms and 2298ms, respectively. However, the prevailing methods attained a high average fuzzification and defuzzification time of 6848.75ms and 6877.75ms, correspondingly.

Table 1.
Rule generation time analysis.

Techniques	Rule generation time (ms)
Proposed ZSH-Fuzzy	3367
Fuzzy	4897
Sigmoid Fuzzy	8543
Trapezoidal Fuzzy	12568
Triangular Fuzzy	16237

Rule generation time analysis of the proposed ZSH-Fuzzy and existing methods is shown in Table 1. The proposed ZSH-Fuzzy achieved a low rule generation time of 3367ms. But, the existing methods obtained high rule generation time.Here, the ZS-shaped hyperbolic membership functionisused to improve the rule generation.

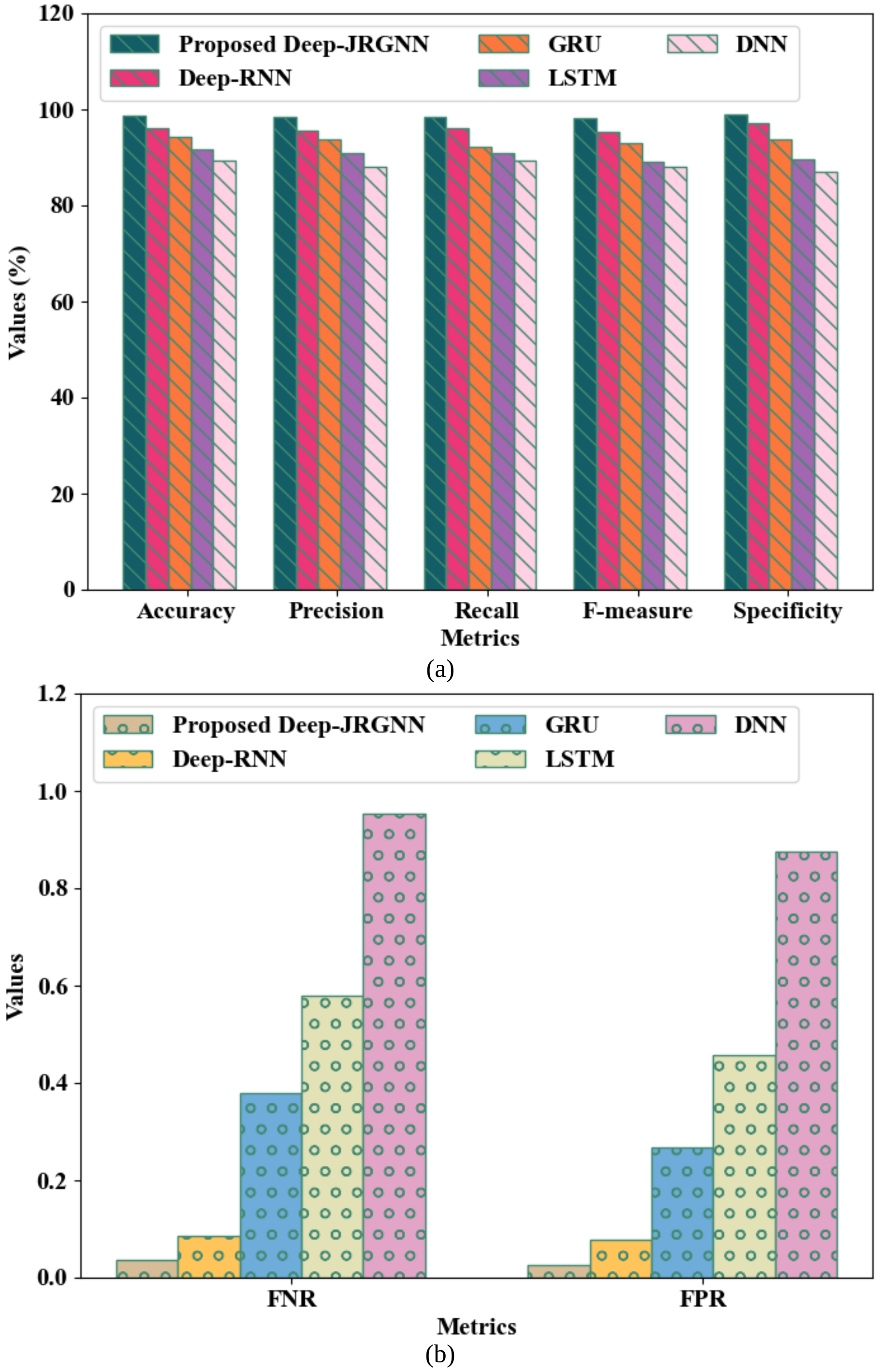


Figure 4.
Comparative evaluation regarding (a) accuracy, precision, recall, F-measure, specificity (b) FNR, FPR.

Figures 4 display the comparative evaluation of the proposed Deep-JRGNN and existing methods regarding the performance metrics. Here, the proposed Deep-JRGNN achieved high accuracy, precision, recall, F-measure, and specificity of 98.74%, 98.34%, 98.56%, 98.15%, and 98.89%, respectively. Also, the proposed technique obtained a low False Negative Rate (FNR) and False Positive Rate (FPR) of 0.03647 and 0.02578, respectively. However, the prevailing techniques like Deep-RNN, Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), and DNN attained poor performance.

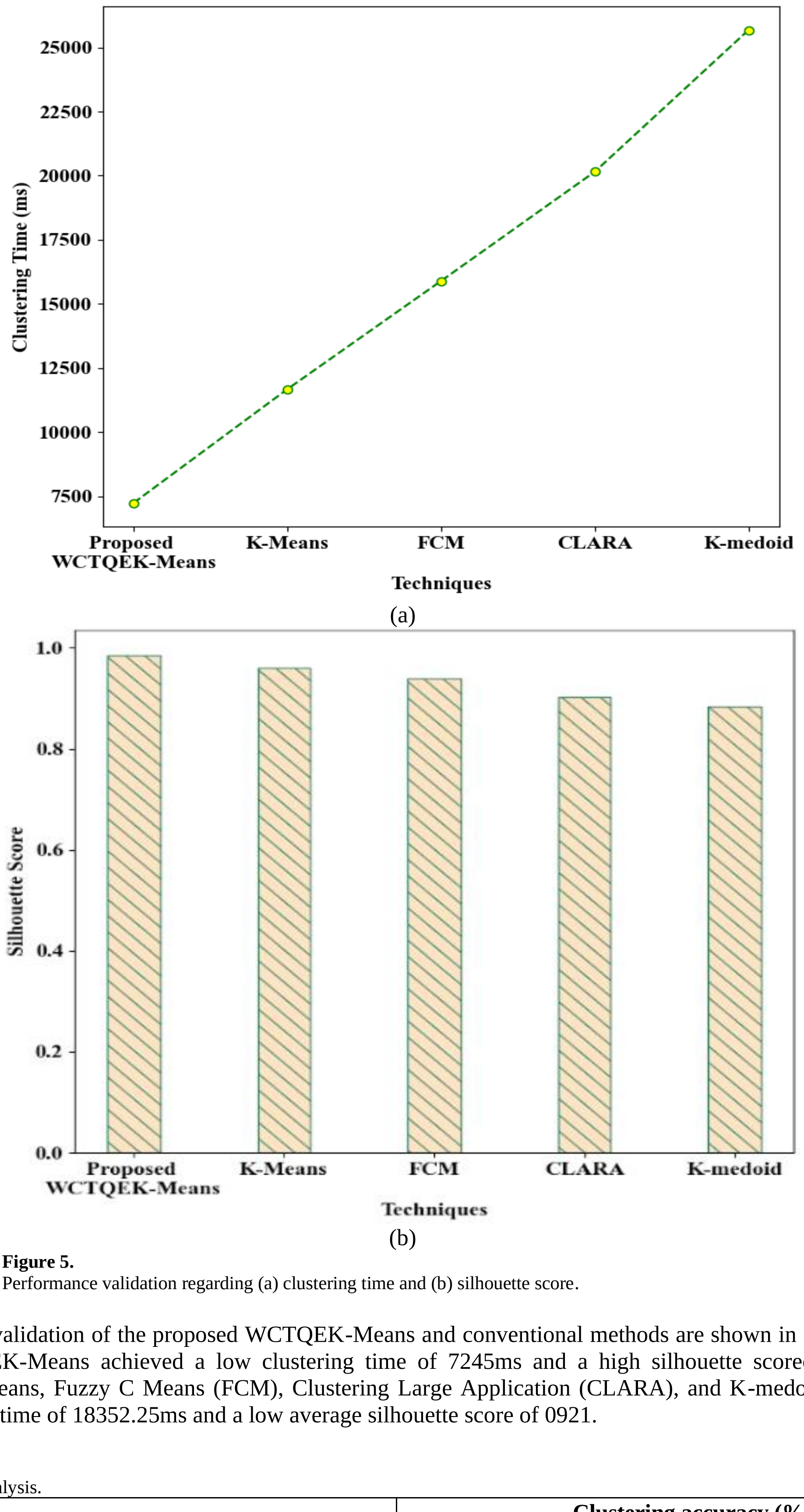


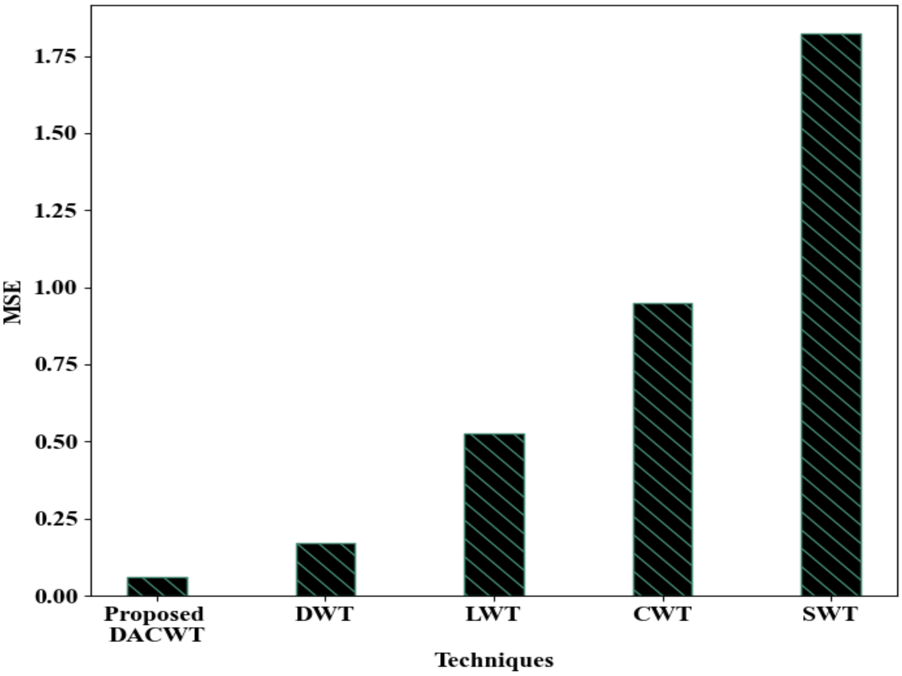
Figure 5.
Performance validation regarding (a) clustering time and (b) silhouette score.

Performance validation of the proposed WCTQEK-Means and conventional methods are shown in Figure 5. Here, the proposed WCTQEK-Means achieved a low clustering time of 7245ms and a high silhouette scoreof 0.986. But, the conventional K-Means, Fuzzy C Means (FCM), Clustering Large Application (CLARA), and K-medoid obtained a high average clustering time of 18352.25ms and a low average silhouette score of 0921.

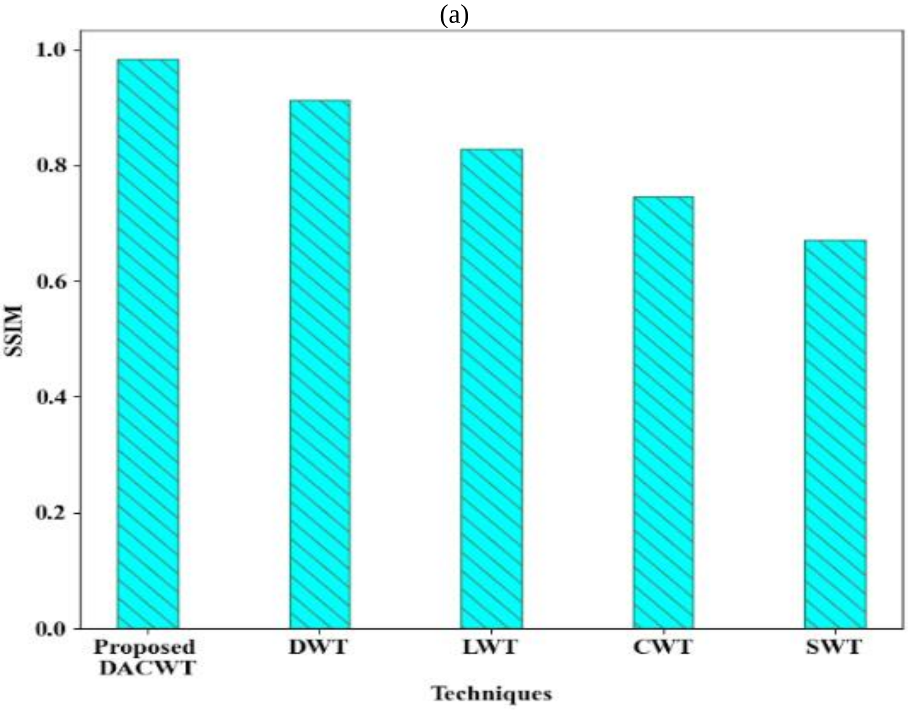
Table 2.
Clustering accuracy analysis.

Techniques	Clustering accuracy (%)
Proposed WCTQEK-Means	98.72
K-means	95.32
FCM	93.89
CLARA	91.45
K-medoid	88.56

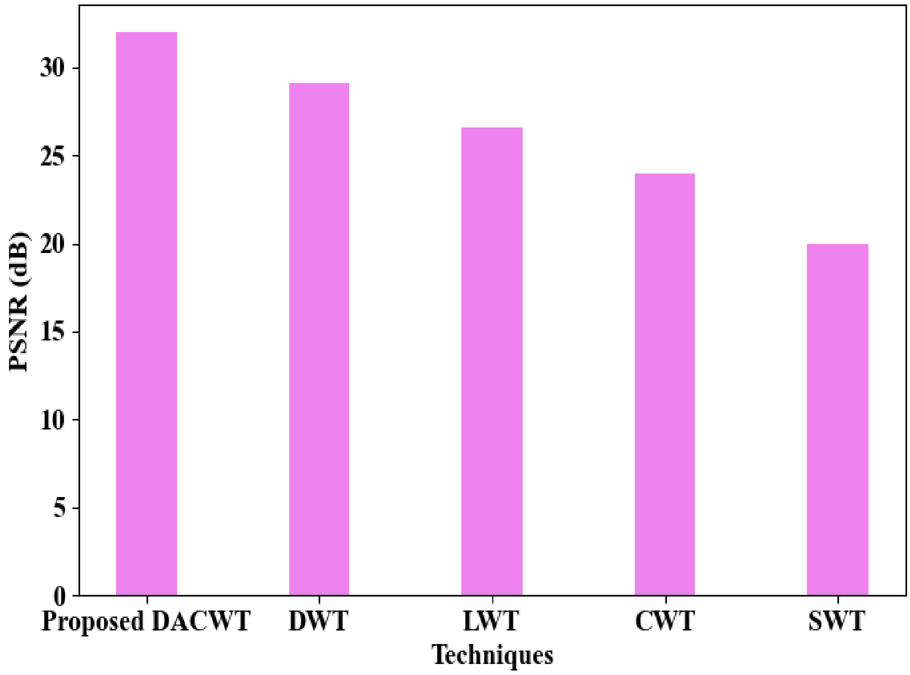
Table 2 displays the clustering accuracy analysis of the proposed model and existing techniques. Here, the proposed WCTQEK-Means obtained a high clustering accuracy of 98.72%, whereas the existing techniques attained low clustering accuracy.



(a)



(b)



(c)

Figure 6.
Performance estimation regarding (a) MSE (b) SSIM (c) PSNR.

Figure 6 depicts the performance estimation of the proposed model and prevailing methods. Here, the proposed DACWT obtained a low Mean Square Error (MSE)of 0.0587; it also attained a high Structural Similarity Index Measure (SSIM)and Peak Signal-to-Noise Ratio (PSNR)of 0.9845 and 31.972 decibels (dB), respectively. Likewise, conventional methods like DWT, Lifting Wavelet Transform (LWT), Continuous Wavelet Transform (CWT), and Stationary Wavelet Transform (SWT) attained poor performance metrics.

4.3. Comparative Analysis

Here, the comparative analysis is done as follows.

Table 3.
Comparative analysis.

Author’s name	Techniques	Accuracy (%)	Precision (%)	Recall (%)
Proposed model	Deep-JRGNN	98.74	98.34	98.56
Nabil, et al. [29]	DNN	89	88	89
Yağcı [30]	Hybrid ML	74	75	74
Yousafzai, et al. [31]	Attention-based BiLSTM	90.16	90	90
Asselman, et al. [32]	XGBoost	78.75	75.12	78.75
Feng, et al. [33]	CNN	94.59	-	-

Table 3 displays the comparative analysis. Here, the proposed Deep-JRGNN obtained a high accuracy, precision, and recall of 98.74%, 98.34%, and 98.56%, respectively. But, the prevailing DNN, Hybrid ML, Attention-based Bidirectional LSTM (BiLSTM), eXtreme Gradient Boosting (XGBoost), and Convolutional Neural Network (CNN) attained poor performance metrics. Thus, the proposed model is better at performing performance modelling.

5. Conclusion

This paper presentedgamification elements-based engineering students’ learning behavior analysis and performance modelling using Deep-JRGNN and ZSH-Fuzzy. Here, the “Student Performance Prediction Dataset” was employed to assess the proposed model. The proposed Deep-JRGNN obtained a high accuracy, precision, and recall of 98.74%, 98.34%, and 98.56%, respectively. Likewise, the proposed ZSH-Fuzzy took a less time of 3367ms for rule generation. Thus, the proposed model accurately predicted the engineering student’s learning behavior. However, the model only concentrated on performance modelling; also, it failed to ensure the security of authorized student’s digital certificates.

5.1. Future Enhancement

In the future, improved cryptography mechanisms will be introduced to secure the students' digital certificates from unauthorized access.

References

[1] G. Feng, M. Fan, and Y. Chen, "Analysis and prediction of students' academic performance based on educational data mining," *IEEE Access,* vol. 10, pp. 19558-19571, 2022. https://doi.org/10.1109/ACCESS.2022.3151652

[2] C. J. Arizmendi *et al.*, "Predicting student outcomes using digital logs of learning behaviors: Review, current standards, and suggestions for future work," *Behavior Research Methods,* vol. 55, pp. 3026-3054, 2023. https://doi.org/10.3758/s13428-022-01939-9

[3] D. Hooshyar and Y. Yang, "[Retracted] predicting course grade through comprehensive modelling of students' learning behavioral pattern," *Complexity,* vol. 2021, no. 1, p. 7463631, 2021. https://doi.org/10.1155/2021/7463631

[4] H. Pallathadka, A. Wenda, E. Ramirez-Asís, M. Asís-López, J. Flores-Albornoz, and K. Phasinam, "Classification and prediction of student performance data using various machine learning algorithms," *Materials Today: Proceedings,* vol. 80, pp. 3782-3785, 2023. https://doi.org/10.1016/j.matpr.2021.07.382

[5] C. F. Rodríguez-Hernández, M. Musso, E. Kyndt, and E. Cascallar, "Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation," *Computers and Education: Artificial Intelligence,* vol. 2, p. 100018, 2021. https://doi.org/10.1016/j.caeai.2021.100018

[6] B. Albreiki, N. Zaki, and H. Alashwal, "A systematic literature review of student' performance prediction using machine learning techniques," *Education Sciences,* vol. 11, no. 9, p. 552, 2021. https://doi.org/10.3390/educsci11090552

[7] Y. Baashar *et al.*, "Toward predicting student's academic performance using artificial neural networks (ANNs)," *Applied Sciences,* vol. 12, no. 3, p. 1289, 2022. https://doi.org/10.3390/app12031289

[8] S. Albahli, "Efficient hyperparameter tuning for predicting student performance with Bayesian optimization," *Multimedia Tools and Applications,* vol. 83, pp. 52711-52735, 2024. https://doi.org/10.1007/s11042-023-17525-w

[9] Y. Baashar, G. Alkawsi, N. a. Ali, H. Alhussian, and H. T. Bahbouh, "Predicting student's performance using machine learning methods: A systematic literature review," in *2021 International Conference on Computer & Information Sciences (ICCOINS) (pp. 357-362). IEEE*, 2021.

[10] P. Jiao, F. Ouyang, Q. Zhang, and A. H. Alavi, "Artificial intelligence-enabled prediction model of student academic performance in online engineering education," *Artificial Intelligence Review,* vol. 55, pp. 6321-6344, 2022. https://doi.org/10.1007/s10462-022-10155-y

[11] S. Li and T. Liu, "Performance prediction for higher education students using deep learning," *Complexity,* vol. 2021, no. 1, p. 9958203, 2021. https://doi.org/10.1155/2021/9958203

[12] Z. Kanetaki *et al.*, "Grade prediction modeling in hybrid learning environments for sustainable engineering education," *Sustainability,* vol. 14, no. 9, p. 5205, 2022. https://doi.org/10.3390/su14095205

[13] S. A. Priyambada, M. Er, B. N. Yahya, and T. Usagawa, "Profile-based cluster evolution analysis: Identification of migration patterns for understanding student learning behavior," *IEEE Access,* vol. 9, pp. 101718-101728, 2021. https://doi.org/10.1109/ACCESS.2021.3095958

[14] H. Ayaz *et al.*, "Optical imaging and spectroscopy for the study of the human brain: Status report," *Neurophotonics,* vol. 9, no. S2, p. S24001, 2022. https://doi.org/10.1117/1.NPh.9.S2.S24001

[15] Y. Luo, Z. Ye, and R. Lyu, "Detecting student depression on Weibo based on various multimodal fusion methods," in *Fourth International Conference on Signal Processing and Machine Learning (CONF-SPML 2024) (Vol. 13077, pp. 202-207). SPIE,* 2024.

[16] S. Leelaluk *et al.*, "Attention-based artificial neural network for student performance prediction based on learning activities," *IEEE Access,* vol. 12, pp. 100659-100675, 2024. https://doi.org/10.1109/ACCESS.2024.3429554

[17] F. Giannakas, C. Troussas, I. Voyiatzis, and C. Sgouropoulou, "A deep learning classification framework for early prediction of team-based academic performance," *Applied Soft Computing,* vol. 106, p. 107355, 2021. https://doi.org/10.1016/j.asoc.2021.107355

[18] N. Massa, M. Dischino, J. Donnelly, and F. Hanes, "Problem-based learning in photonics technology education: Assessing student learning," in *Education and Training in Optics and Photonics (p. ETB4). Optica Publishing Group*, 2009.

[19] M. Firoz, M. M. Islam, M. Shidujaman, A. Islam, and M. T. Habib, "University student's mental stress detection using machine learning," in *Seventh International Conference on Mechatronics and Intelligent Robotics (ICMIR 2023) (Vol. 12779, pp. 757-767). SPIE,* 2023.

[20] J. T. H. Smith, S. W. Linderman, and D. Sussillo, "Reverse engineering recurrent neural networks with Jacobian switching linear dynamical systems," in *Advances in Neural Information Processing Systems, 34, 16,700–16,713. Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS '21). Curran Associates, Inc*, 2021.

[21] I. K. Fodor and C. Kamath, "Denoising through wavelet shrinkage: An empirical study," *Journal of Electronic Imaging,* vol. 12, no. 1, pp. 151-160, 2003. https://doi.org/10.1117/1.1525793

[22] Y. Wang, Z. Li, C. Wang, L. Feng, and Z. Zhang, "Implementation of discrete wavelet transform," in *2014 12th IEEE International Conference on solid-state and integrated circuit technology (ICSICT) (pp. 1-3). IEEE,* 2014.

[23] M. D. S. Lubis, H. Mawengkang, and S. Suwilo, "Performance analysis of entropy methods on K means in clustering process," in *Journal of Physics: Conference Series (Vol. 930, No. 1, p. 012028). IOP Publishing*, 2017.

[24] N. B. Kumarakulasinghe, T. Blomberg, J. Liu, A. S. Leao, and P. Papapetrou, "Evaluating local interpretable model-agnostic explanations on clinical machine learning classification models," in *2020 IEEE 33rd international symposium on computer-based medical systems (CBMS) (pp. 7-12). IEEE,* 2020.

[25] D. Kurniadi, A. Mulyani, and I. Muliana, "Prediction system for problem students using k-nearest neighbor and strength and difficulties questionnaire," *Jurnal Online Informatika,* vol. 6, no. 1, pp. 53-62, 2021.

[26] M. A. Marjan, M. P. Uddin, and M. Ibn Afjal, "An educational data mining system for predicting and enhancing tertiary students' programming skill," *The Computer Journal,* vol. 66, no. 5, pp. 1083-1101, 2023. https://doi.org/10.1093/comjnl/bxab214

[27] S. D. A. Bujang *et al.*, "Multiclass prediction model for student grade prediction using machine learning," *Ieee Access,* vol. 9, pp. 95608-95621, 2021. https://doi.org/10.1109/ACCESS.2021.3093563

[28] X. Chen, M. Vorvoreanu, and K. Madhavan, "Mining social media data for understanding students' learning experiences," *IEEE Transactions on Learning Technologies,* vol. 7, no. 3, pp. 246-259, 2014. https://doi.org/10.1109/TLT.2013.2296520

[29] A. Nabil, M. Seyam, and A. Abou-Elfetouh, "Prediction of students' academic performance based on courses' grades using deep neural networks," *IEEE Access*, vol. 9, pp. 140731-140746, 2021. https://doi.org/10.1109/ACCESS.2021.3119596

[30] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learning Environments*, vol. 9, p. 11, 2022. https://doi.org/10.1186/s40561-022-00192-z

[31] B. K. Yousafzai *et al.*, "Student-performulator: Student academic performance using hybrid deep neural network," *Sustainability*, vol. 13, no. 17, p. 9775, 2021. https://doi.org/10.3390/su13179775

[32] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3360-3379, 2023. https://doi.org/10.1080/10494820.2021.1928235

[33] G. Feng, M. Fan, and C. Ao, "Exploration and visualization of learning behavior patterns from the perspective of educational process mining," *IEEE Access*, vol. 10, pp. 65271-65283, 2022. https://doi.org/10.1109/ACCESS.2022.3184111