Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, Special Issue, 2025

" Emerging Trends, Challenges, and Innovative Practices in Education"

Leveraging Hybrid Deep Q-Learning for Early Identification of At-Risk Students

P. Vijaya [1*], Joseph Mani [1], Basant Kumar [1], Hothefa Shaker [1], Rajeev Rajendran [1]

[1] *Department of Mathematics and Computer Science, Modern College of Business and Science, Bowshar, Muscat, Sultanate of Oman.*

Abstract

Student performance prediction is employed to predict the learning performance to identify at-risk students. However, prediction models should also consider external factors along with learning activities, such as course duration. The student's performance gets affected, which leads to a high decreasing rate and meets the risk of failing to complete the course on time. To overcome these challenges, this work proposed a Sea Lion Search Optimization (SLnSO) based on the Deep Q network (DQN) for predicting at-risk students. Here, the input data is taken from the dataset and forwarded to the data transformation phase, which is performed by Yeo-Johnson (YJ) transformation. Then, in the feature selection stage, the most relevant features are selected using the Damerau-Levenshtein technique. Then, Data Augmentation (DA) is performed to increase the dimension of the features, which is followed by the Deep Q Network (DQN) that is utilized for predicting the students at risk. Finally, by implementing the proposed SLnSO, the predicted results will be executed by DQN. The SLnSO-DQN is the combination of both Sea Lion Optimization (SLnO) and Squirrel Search Algorithm (SSA). The outcomes of the proposed model SLnSO-DQN attain significant performance that is based on various parameters, such as Mean Absolute Error (MAE), Mean Square Error (MSE), and Root MSE (RMSE), and also obtained better values of 0.327, 0.265, and 0.514, respectively.

Keywords:

Student Performance Prediction;
Data Mining;
Deep Learning;
At-Risk Students;
Optimization;
Deep Q Network.

Article History:

Received:	05	February	2025
Revised:	15	June	2025
Accepted:	04	July	2025
Published:	12	August	2025

1- Introduction

Considering the field of big data with fast progress, there has been growing importance of effectual data analytic tools like Data Mining (DM) for determining essential information by extracting the data from the dataset. DM is a highly significant process in data analysis, as the efficiency of the process depends on the in-depth, intelligent, and most relevant data extracted [1]. One of the subsets of DM, named Educational Data Mining (EDM), has enormous commercial opportunities and produces effective results in the educational domain by enhancing the quality of education provided by analyzing the learning datasets [2]. EDM is a developing transdisciplinary research area that handles the creation of statistical and computational strategies to evaluate and investigate educational information. In practice, EDM permits identifying the information depending on students' learning data for endorsing and evaluating the educational system, resulting in enhanced quality of education [3]. In recent years, there have been increasing instances of students leaving school when forced or warned to excel academically [4].

Student performance prediction is a highly significant process in the creation and progress of any online learning platform [5]. It estimates if the students can respond accurately to the questions in the future [6]. It further intends to recognize the student's learning capability or knowledge expertise [7]. An effective approach can minimize the student's

* **CONTACT**: vijaya.padmanabha@mcbs.edu.om

DOI: http://dx.doi.org/10.28991/ESJ-2025-SIED1-06

learning difficulties by providing tailored learning programs and reduce the energy and time of the educators by offering instantaneous feedback [8, 9]. Multiple definitions exist for defining student performance, such as in Bin Mat et al. [10]; it can be achieved by calculating the education program and the evaluation. Nevertheless, numerous types of research have indicated that the success of any student depends on graduating [11, 12]. A major cause for the students performing poorly is that they have not chosen suitable programs matching their capabilities, resulting in a delay in course completion, thus causing an enhanced cost for society, higher educational institutions, and families [4, 13]. Consequently, estimating the student's performance is a major research area in the educational domain, with several works addressing the issue [14-16].

Recently, student performance prediction has been carried out using various approaches, like clustering, regression, classification, etc., among which classification is considered the most common. Classifiers, like K-nearest neighbor (KNN), linear regression (LinR), Support Vector Machine (SVM), Artificial Neural Network (ANN), Naive Bayes (NB), and Decision Tree (DT) have been employed for predicting student performance [17]. The statement "at-risk students" normally points out the students who find difficulty while studying, are prone to fail in academics, or have an increased tendency to drop out [18]. A surge in the number of "at-risk students" contributes to the rise in dropout rates and incomplete courses, thereby leading to low education quality, impacting the concerned stakeholders such as senior management, academic counsellors, tutors, and students [19, 20]. The above-discussed issue can be addressed by efficient alerting strategies based on DM for identifying the indicators and utilizing them for recognizing the at-risk students and providing them adequate support [18]. In recent years, Machine Learning (ML) has been commonly used in different techniques [21-23] to recognize at-risk students who cannot complete their learning program on time [24, 25]. However, the application of the ML techniques approaches is limited by the smaller number of input variables [26]. The progress in deep learning has produced highly efficient predictions, owing to its dimensionality reduction and feature extraction capabilities [9].

1-1- Motivation

Identifying at-risk students and improving academic success is very important for student performance prediction. The early identification of at-risk students improves academic outcomes. There are many prediction methods have been developed already, and the limitations of those methods are listed below:

- In many traditional methods, the identification of at-risk students is done based on internal learning behaviors, such as attendance, assignments, and quiz scores, without considering external factors like course duration. It leads to incomplete assessments and reduces the overall performance.
- The accuracy of some existing models depends on the feature selection techniques. The traditional feature selection approaches may discard important features and it leads to the accuracy reduction.
- The raw data used in the existing methods contains inconsistencies, missing values and skewed distributions, which affected the learning ability of the existing deep learning approaches.
- The traditional approaches performed well on training data and failed to generalize to new student records.
- The traditional machine learning models mainly rely on static optimization techniques and they failed to capture the dynamic nature of student performance trends over time.

These limitations in the existing methods are considered as a motivation for developing a new model, named SLnSO-DQN for at-risk student prediction. In the proposed SLnSO-DQN method, the course duration is considered a key external factor in the prediction model, which permits a more holistic assessment of student performance. Also, the feature selected is conducted by the Damerau-Levenshtein technique, which finds the most relevant features by determining edit distances between feature values, which ensures that only the most meaningful attributes contribute to prediction. Moreover, for better model learning, the YJ transformation is applied to normalize and standardize data, which reduces the skewness and improves feature distribution. Furthermore, the feature diversity is enhanced by applying the DA, which makes the approach more robust in handling various student learning patterns. Additionally, in this research, the SLnO and SSA are hybridized for better hyperparameter tuning of DQN, which ensures adaptive and accurate student performance predictions.

In this work, an efficient at-risk student prediction technique using SLnSO-DQN is presented. The key contribution of the research work is given below.

- In this research, the SLnSO-based DQN is proposed for predicting the at-risk students.
- Here, the most relevant features are extracted using the Damerau-Levenshtein (DL) and the SLnSO.
- The DQN is used for the prediction of at-risk students, by which the proposed SLnSO algorithm is used for training the DQN.
- The SLnSO is formulated by utilizing the SLnO algorithm and SSA.

The remaining part of this paper is organized in the following manner: Section 2 presents the literature review. In section 3, the proposed SLnSO-DQN is explained and section 4 describes the assessment of the experimental results. Finally, the conclusion of the work is presented in section 5.

2- Literature Review

The issues of at-risk students were studied by numerous works and performance predictions; among them, only twelve are considered in this research. Azcona et al. [27] developed an ML algorithm for automatically identifying at-risk students in computer programming classes. Here, a predictive student model was built using static details, like activity logs, demographics, and characteristics of each student. Further, it provided weekly predictions for the instructors and students during the semester. This technique effectively minimizes the gap between the best and poor-performing students. However, it failed to extract the usage patterns from more student groups. The usage pattern was extracted from the study by Susheelamma & Ravikumar [25], in which they introduced a novel XGBoost-based classification algorithm (PXGB) for identifying at-risk students, without relying on historical academic data. The approach identified the student's behaviour based on the tasks submitted and then identified the pattern. Once the pattern was identified, the at-risk students were identified using the XGBoost classifier. This technique effectively identified the duration of course completion with high accuracy, but it was unsuccessful in utilizing various datasets to enhance the performance. Son & Fujita [28] utilized multiple datasets to develop a technique called Multi Adaptive Neuro-Fuzzy Inference System with Representative Sets (MANFIS-S), aimed at addressing the challenge of predicting student academic performance in a multi-input, multi-output (MIMO) context. This scheme utilized two learning approaches, namely global and local. Global training was used to correct and attain a meaningful parameter subset, while the local training employed two parameters: consequent and premise. Here, a hybrid Particle Swarm Optimization (PSO) and gradient descent were employed for training the parameters.

Finally, classification was performed using the Fuzzy K-Nearest Neighbour. The MANFIS-S was effective in attaining high accuracy, although it did not consider the examination of the convergence rate. He et al. [2] explored the convergence rate by introducing a joint Recurrent Neural Network (RNN)–Gated Recurrent Unit (GRU) architecture designed to forecast student performance. By considering the student's demographic details and interaction stream data, the at-risk students were determined by utilizing the RNN-GRU. Further, a data completion approach was considered for handling the missing data. This scheme has a high convergence speed, but this technique suffers from high computational complexity. To overcome the drawback in He et al. [2], Kustitskaya et al. [29] developed a Bayesian Network Classifier (BNC) to detect at-risk students. Here, the prediction was done by developing a Bayesian network with the number of study weeks as the number of networks and forecasting was done every week. This methodology effectively identified the at-risk students whose learning behaviour matched that of the failed students; however, the BNC overestimated the chance of student failure.

Dien et al. [4] addressed this issue by devising a deep learning-based approach to forecast student performance. To estimate academic outcomes in the subsequent semester based on prior results, they developed a prediction scheme employing Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) architectures on one-dimensional (1D) data, referred to as CN1D. The technique successfully provided enhanced prediction performance, specifically while employing data transformation. However, to determine the test data's features, the technique of Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) failed to gather sufficient information in the training data. Liu et al. [9] collected sufficient training data to propose a Deep Knowledge Tracing model enhanced by a Multiple Features Fusion Attention mechanism (MFA-DKT), aimed at estimating student performance. Here, the exercise and student behavioural features were employed to identify the performance. The features from the online learning platform were integrated into the knowledge tracing system, and then an ML approach was used to deal with the features. Later, the student's knowledge traces were identified using the RNN, and finally, the prediction was executed using RNN and attention strategy. This technique attained high accuracy but was unsuccessful in considering the relationship amongst knowledge notions to capture knowledge structure while predicting. Yang et al. [26] addressed this issue by introducing a Convolutional Neural Network (CNN) model designed to identify at-risk students. To extract behavioral features, the model considered students' online activity patterns. The CNN was employed to predict whether a student would pass or fail based on these extracted features. This technique produced a minimum misclassification rate and high sensitivity, although the scheme's performance was not enhanced by considering other variable arrangements.

Leelaluk et al. [30] presented an Attention-Based Artificial Neural Network (Attn-ANN) for predicting students' performance based on learning activities. This method provided better accuracy than the designed algorithms. However, the decisions made by the Attn-ANN regarding the actual classes differed and related to another factor that was not included in the prediction model. Mehdi & Nachouki [31] developed a Neuro-Fuzzy Model for predicting and analysing student graduation performance in computing programs. This method produced improved and good predictive results compared to the traditional Multi Linear Regression (MLR) technique based on a comparative analysis. Although the model provided good predictive results, the results of academic program reviews, changes in course content, teaching methodology, and assessment methods caused deviations in grades that affected students' GPA. Ouyang et al. [32] developed an AI performance prediction model using learning analytics (LA) approaches to improve student learning effects. The AI performance prediction model obtained high effectiveness in the LA technique. However, it failed to expand the educational contexts, course subjects, as well as sample size to test and verify the empirical research results and implications.

Kusumawardani & Alfarozi [33] developed a Transformer Encoder Model (TEM) method for predicting student performance from log activities. This technique provided successful enhanced performance prediction. Though the data collection mechanism was integrated with the Learning Management System (LMS), this method cannot be applied to all LMSs that provide a data architecture similar to OULAD. Noviandy et al. [34] established a stacked classifier, which merges the advantages of logistic regression, Random Forest, and LightGBM for dropout risk prediction. The predictive capability of the model was high and it achieved academic excellence early. However, the ensemble classifier required more computational time and resources. Awedh & Mueen [35] established a hybrid Logistic Regression-K-Nearest Neighbour (LR-KNN) scheme for improving student achievement and minimizing the number of dropouts. Here, a new feature extraction approach, named Genetic Algorithm-optimized Latent Dirichlet Allocation (GA-LDA) was implemented for effective feature extraction and accurate student grouping was done by Gaussian Flow Optimizer (GFO). However, the model was trained on a limited dataset, so that overfitting issue was possible.

3- Proposed Sea Lion Search Optimization – Deep Q Network for at-risk student prediction

This section deliberates the presented SLnSO-DQN for predicting at-risk students. In any educational system, predicting students' performance is highly challenging, as it needs numerous details regarding the student for accurate prediction. The developed SLnSO-DQN approach for predicting at-risk students is realized from the phases that include data acquisition, data transformation, feature section, data augmentation, and student at-risk prediction. At first, the input is gathered based on the student's behaviour from the dataset, followed by the data transformation. Here, data transformation is performed using Yeo-Johnson transformation (YJT) [36] to convert the data into a standard format for further processing. The feature selection stage is fed by the transformed data, wherein the SLnSO algorithm chooses the salient features, based on DL distance as the fitness. Here, the developed SLnSO algorithm is devised by modifying the SSA [37] with respect to the SLnO [38]. After the features are selected, data augmentation is executed to enhance the data dimensionality. Finally, the student performance prediction is computed to find at-risk students using DQN [39], whose weights and biases are updated using the devised SLnSO algorithm. The schematic view of the developed SLnSO-DQN for the prediction of at-risk students is shown in Figure 1. Using the SLnSO-DQN, the process of predicting at-risk students is explained in the succeeding subsections.

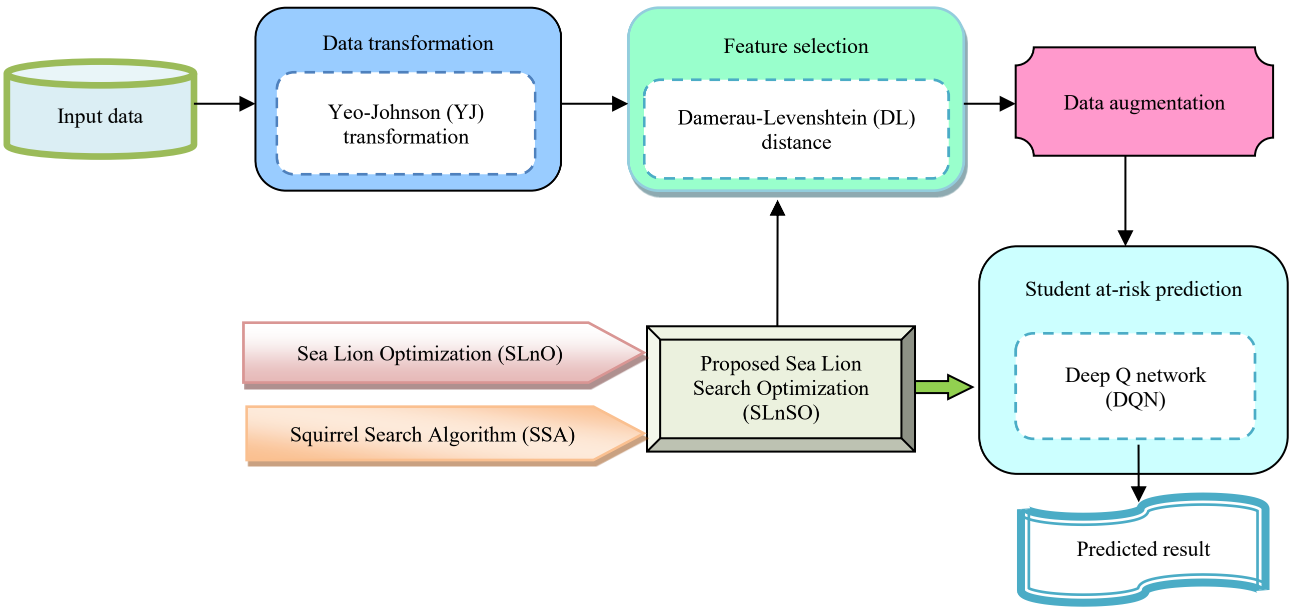


Figure 1. Schematic view of the developed SLnSO-DQN for predicting at-risk students

3-1- Data Acquisition using Student Performance Prediction Dataset

The prediction of at-risk students is carried out using the dataset P that comprises of P number of data concerning various attributes of the students. The dataset with several attributes is represented by:

$$P = \{P_1, P_2, \dots, P_i, \dots, P_p\} \tag{1}$$

here, the dimension of data is $u \times v$ and P_i conveys the i^{th} data in the database. Here various attributes of students from the dataset include school, sex, age, address, guardian, study time, failures, absence, health and so on [40].

3-2- Data transformation using Yeo-Johnson (YJ) transformation

P_i is exposed to the data transformation procedure, which is pre-processed using YJT. Data transformations are used to convert the raw information into a structured format to enhance the interpretability of the data. YJT is a Box-Cox transformation (BCT) technique that can be applied to any numeric data, irrespective of data is positive or negative. YJT is a nonparametric transformation that combines the transformations for various data into a single parameter using smoothness criteria. The following expression gives the YJT.

$$\lambda(P_i) = \begin{cases} \frac{(P_i+1)^{\alpha}-1}{\alpha}, & \alpha \neq 0 \; and \; P_i \geq 0 \\ ln(P_i+1), & \alpha = 0 \; and \; P_i \geq 0 \\ \frac{-((-P_i+1)^{2-\alpha}-1)}{2-\alpha}, & \alpha \neq 2 \; and \; P_i < 0 \\ ln(-P_i+1), & \alpha = 2 \; and \; P_i < 0 \end{cases} \tag{2}$$

here, α is the index parameter, $\lambda(P_i)$ gives the YJT of the data P_i and is then applied to the feature selection step.

3-3- Feature selection using the Developed Sea Lion Search Optimization (SLnSO) Algorithm

The processed data $\lambda(P_i)$ is used for the selection of the most salient features in the input data to reduce the data dimensionality. Here, the input data of dimension $u \times v$ is minimized to a dimension of $u \times w$, with $v > w$. Feature selection is performed using the proposed SLnSO algorithm, which is developed by changing the location of the squirrel in SSA [37] in accordance with SLnO [38]. Among various features available in the dataset, some of the features like school, sex, fam size, P status and guardian are selected for further processing. The feature selection process is explained below.

3-3-1- Solution Encoding

Feature selection aims to select the salient features from the data that give an accurate representation of the data. The pictorial illustration of the solution is solution encoding for the feature selection problem, wherein each solution corresponds to the features. The solution encoding of the feature selection process is represented in Figure 2. Here, $1 \times a$ denotes the total features to be selected, $1,2,3,\ldots.,a$ denotes the number of features, and the corresponding index of every feature is given inside the boxes. The most significant features are selected based on the fitness function, which is explained in Section 3.3.2.

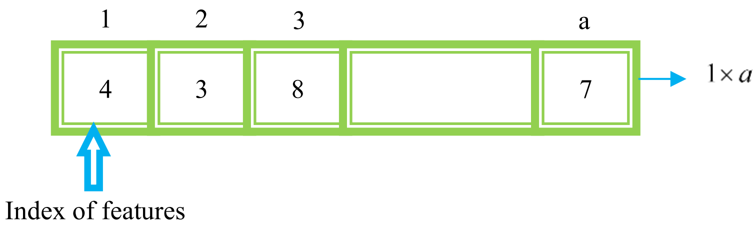


Figure 2. Solution encoding

3-3-2- Fitness Function

The most salient features are selected by considering the DL distance [41] as the fitness measure. The most significant features corresponding to the minimum DL distance, which is expressed as,

$$DL_{x,y}(p,q) = min \begin{cases} 0 & if \; p = q = 0 \\ DL_{x,y}(p-1,q)+1 & if \; p > 0 \\ DL_{x,y}(p,q-1)+1 & if \; q > 0 \\ DL_{x,y}(p-1,q-1)+1_{(x_p \neq y_q)} & if \; p,q > 0 \\ DL_{x,y}(p-2,q-2)+1 & if p,q > 1 \; and \; x_p = y_{q-1} \; and \; x_{p-1} = y_q \end{cases} \tag{3}$$

here, x signifies the aspirant feature and $y$is the target feature, $p$and $q$represents the row and column respectively. When $p > 0$ and $q = 0$, the corresponding candidate feature is neglected, and while $p = 0$ and $q > 0$, the candidate feature is replaced by any random feature. If both $p,q > 0$, then the features are considered to be similar or dissimilar, and the last condition indicates transposition.

3-3-3- Proposed SLnSO Algorithm for Feature Selection

This section elucidates the introduced SLnSO algorithm for selecting the input data's most relevant features. The proposed SLnSO algorithm is devised by adapting the locations of the squirrel in SSA [37] as per the SLnO [38] algorithm. The SSA is an optimization algorithm founded by considering the constantly changing foraging nature of the southern flying squirrels, wherein the squirrels use a technique for travelling called gliding, normally utilized by tiny mammals to cover larger distances. When the weather is warm, the squirrels glide from one tree to another and keep changing their positions to find food. They meet their everyday energy requirements using acorn nuts that are easily obtained. Later, they search for the optimal source of food (hickory nuts) to be saved for winter. The squirrels satisfy their energy needs using the hickory nuts in the winter. Once the winter ends, the squirrels become active again, and the process is carried out during the entire life of the squirrel. The SSA offers the advantage of high search space exploration; however, it failed to determine solutions to NP-hard combinatorial optimization issues. The hunting pattern of the sea lion is seen in the SLnO algorithm. Further, it is inspired by the prey detection process of sea lions using their whiskers. The important stages in the sea lions' hunting process are a) tracing and pursuing the prey utilizing their whiskers, b) encircling the prey by inviting other sea lions into their group, and c) attacking the target. The SLnO is effective in attaining high convergence and exploration. Hence, by merging the SSA and SLnO algorithms, the proposed SLnSO algorithm achieves a high convergence rate. SLnSO algorithm procedures are elaborated as follows.

Step 1) Population Initialization:

The search space is initiated with a population of flying squirrels, and their position is articulated using the Equation given below,

$$Z = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \cdots & \cdots & Z_{1,b} \\ Z_{2,1} & Z_{2,2} & \cdots & \cdots & Z_{2,b} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Z_{c,1} & Z_{c,2} & \cdots & \cdots & Z_{c,b} \end{bmatrix} \tag{4}$$

here, c denotes the number of flying squirrels, b represents the number of decision variables, $Z_{j,k}$ designates the position of the j^{th}squirrel in the k^{th}dimension and the initial location is given by,

$$Z_j = Z_{low} + B(0,1) \times \left(Z_{up} - Z_{low}\right) \tag{5}$$

where, $B(0,1)$ is an arbitrary number dispersed homogeneously in the range $[0,1]$ Z_{up} and Z_{low} indicates the upper and lower limits of thej^{th} flying squirrel in the k^{th} dimension.

Step 2) Fitness evaluation:

Once the location of the flying squirrels is initiated, the fitness of all squirrels is computed using Equation 3, wherein the optimal solution correlates to the squirrel with minimal fitness.

Step 3) Sorting and random selection:

After computing the fitness of the squirrels, the squirrels are sequenced by considering the increasing fitness order. The flying squirrel with low fitness is contemplated to be on a hickory nut tree, wherein three successive flying squirrels are on acorn nut trees, and the remaining are on ordinary trees. After fulfilling their everyday energy requirements, the squirrels present in acorn trees will move onto the hickory trees. Likewise, the ones on the ordinary trees advance to the acorn trees to satisfy their energy requirements. In certain cases, the squirrels in normal trees with satisfied daily requirements will advance to the hickory trees. The prevalence of predators impacts the squirrels' food-searching pattern, which is represented by considering the probability of predator presence P_{qr}.

Step 4) Produce new positions through gliding:

The flying squirrels glide to their next location based on the three scenarios discussed above. These gliding behaviours depend on the presence of the predator. If any predator is present nearby, the squirrel makes cautious steps toward any arbitrary location, modelled by the equation given below.

Scenario 1): In this instance, the squirrels on the acorn tree advance toward the hickory tree to collect the hickory nuts for storage, and this behaviour is expressed as,

$$Z_{ac}^{t+1} = \begin{cases} Z_{ac}^{t} + g_d \times Gc \times (Z_{hc}^{t} - Z_{ac}^{t}), & C_1 \geq P_{qr} \\ Randomposiiton, & otherwise \end{cases} \tag{6}$$

here, Z_{ac}^{t} and Z_{hc}^{t} characterize the location of the squirrel on the acorn trees and hickory nut tree respectively, t is the present iteration, C_1 represents the random number in the range $[0,1]$, g_d is the gliding distance and Gc denotes the gliding constant with value 1.9.

Scenario 2): Here, the movement of the squirrels to the acorn trees from normal trees to satisfy their everyday energy requirements. The location of the squirrels is upgraded using Equation 7.

$$Z_{nor}^{t+1} = \begin{cases} Z_{nor}^{t} + g_d \times Gc \times (Z_{ac}^{t} - Z_{nor}^{t}), & C_2 \geq P_{qr} \\ Randomposiiton, & otherwise \end{cases} \tag{7}$$

where, Z_{nor}^{t} designates the location of squirrels on normal trees and C_2 is the arbitrary number with values between 0 and 1.

Now consider, $Z_{nor}^{t+1} = Z(t+1)$, $Z_{nor}^{t} = Z(t)$, and $Z_{ac}^{t} = Z_{ac}(t)$, and for $C_2 \geq P_{qr}$, then we get,

$$Z(t+1) = Z(t) + g_d \times Gc \times \left(Z_{ac}(t) - Z(t)\right) \tag{8}$$

$$Z(t+1) = Z(t) + g_d \times Gc \times Z_{ac}(t) - g_d \times Gc \times Z(t) \tag{9}$$

$$Z(t+1) = Z(t)(1 - g_d \times Gc) + g_d \times Gc \times Z_{ac}(t) \tag{10}$$

From the SLnO [38] algorithm, the encircling behaviour of the sea lions while chasing and hunting a bait of fish is given by,

$$\vec{S}_l(t+1) = \left|\vec{V}(t) - \vec{S}_l(t)\right| cos(2\pi z) + \vec{V}(t) \tag{11}$$

here, $\vec{V}(t)$and $\vec{S}_l(t)$designates the location of the victim and sea lion, respectively and z denotes a random number with values in $[-1,1]$. Assuming, $\vec{V}(t) > \vec{S}_l(t)$, $\vec{S}_l(t+1) = Z(t+1)$, $\vec{V}(t) = V(t)$and$\vec{S}_l(t) = Z(t)$, the above expression can be rephrased as,

$$Z(t+1) = \left(V(t) - Z(t)\right) cos(2\pi z) + V(t) \tag{12}$$

$$Z(t+1) = V(t)\, cos(2\pi z) - Z(t)\, cos(2\pi z) + V(t) \tag{13}$$

$$Z(t+1) = V(t)(cos(2\pi z) + 1) - Z(t)\, cos(2\pi z) \tag{14}$$

$$Z(t) = \frac{V(t)(cos(2\pi z)+1) - Z(t+1)}{cos(2\pi z)} \tag{15}$$

Substituting Equation 15 in Equation 10,

$$Z(t+1) = \left(\frac{V(t)(cos(2\pi z)+1) - Z(t+1)}{cos(2\pi z)}\right)(1 - g_d \times Gc) + g_d \times Gc \times Z_{ac}(t) \tag{16}$$

$$Z(t+1) + \frac{Z(t+1)(1-g_d \times Gc)}{cos(2\pi z)} = \left(\frac{V(t)(cos(2\pi z)+1)}{cos(2\pi z)}\right)(1 - g_d \times Gc) + g_d \times Gc \times Z_{ac}(t) \tag{17}$$

$$\frac{Z(t+1)(cos(2\pi z)+1-g_d \times Gc)}{cos(2\pi z)} = \frac{V(t)(cos(2\pi z)+1)(1-g_d \times Gc) + g_d \times Gc \times Z_{ac}(t) \times cos(2\pi z)}{cos(2\pi z)} \tag{18}$$

$$Z(t+1) = \frac{V(t)(1-g_d \times Gc)(cos(2\pi z)+1) + g_d \times Gc \times Z_{ac}(t) \times cos(2\pi z)}{(cos(2\pi z)+1-g_d \times Gc)} \tag{19}$$

The location and movement of the flying squirrel from the normal tree to the acorn tree is represented in the above equation

Scenario 3): Here, the squirrels on the normal tree approach the hickory nut tree, as they have already fed on the acorn nuts. The position is updated with the help of the equation given below,

$$Z_{nor}^{t+1} = \begin{cases} Z_{nor}^{t} + g_d \times Gc \times (Z_{hc}^{t} - Z_{nor}^{t}), & C_3 \geq P_{qr} \\ Randomposiiton, & otherwise \end{cases} \tag{20}$$

here, C_3 represents the random number with a value between 0 and 1and the value of $P_{qr}$is set as 0.1 in all scenarios.

The gliding distance g_d is expressed as,

$$g_d = \frac{g_h}{tan\,\theta} \tag{21}$$

here, g_h represents the height loss after gliding, and θ indicates the steady state glide angle expressed as,

$$\theta = arctan\left(\frac{lf}{dg}\right) \tag{22}$$

where, lf denotes the lift and dg is the drag encountered by the squirrel while gliding. The total of the drag and lift results in an output force with a magnitude equal to and direction opposite to the weight of the squirrel W. The lift lf and the drag dg are expressed as follows.

$$dg = \frac{1}{2\rho s^2 Adc} \tag{23}$$

$$lf = \frac{1}{2\rho s^2 ALc} \tag{24}$$

here, ρ indicates the density of air with value $1.204\ kgm^{-3}$, A denotes the surface area of the body having value $154\ cm^2$, s represents the speed and has a value of $5.25\ ms^{-1}$ and Lc and dc designates the coefficient of lift and frictional drag, respectively.

Step 5) Evaluate seasonal monitoring criteria:

The squirrels' foraging pattern varies with seasons, as in winter seasons, they become less active. They also tend to lose energy because of the heat loss in winter, and further, they are prone to attacks due to tree cover loss. Once the winter ends, they become active and start foraging. Hence, a seasonal monitoring criterion is required for the algorithm to avoid local optima. A seasonal constant Z_c is employed for seasonal monitoring and is expressed as,

$$Z_c^t = \sqrt{\sum_{i=1}^{b}\left(Z_{at,i}^t - Z_{hc,i}\right)^2} \tag{25}$$

here, $t = 1,2,3$.

Seasonal monitoring is performed by checking the condition $Z_c < Z_{min}$, with Z_{min} indicating the minimum seasonal constant value, expressed by,

$$Z\frac{10E-6}{(365)^{\left(\frac{t}{\left(\frac{T}{2.5}\right)}\right)}}_{min} \tag{26}$$

wherein, T epitomizes the maximal iteration. The value of Z_{min} impact the exploration and exploitation capabilities.

The seasonal monitoring criterion is checked, and if it is true, then the squirrels are located randomly in the forest to start exploration again.

Step 6) Relocation at winter end:

When the winter season ends, the flying squirrels become active again. The squirrels which have managed to survive start foraging for food. This behaviour can be represented as,

$$Z_{nor}^{new} = Z_{low} + Levy(c) \times \left(Z_{up} - Z_{low}\right) \tag{27}$$

here, $Levy(c)$ indicates the levy flight function, which is used to determine the new solutions from the present optimal solution. The levy flight function is computed using the following expression.

$$Levy(c) = 0.01 \times \frac{m_1 \times \beta}{|m_2|^{\frac{1}{\chi}}} \tag{28}$$

here, m_1 and m_2 represent two arbitrary numbers randomly distributed in the range $[0,1]$ and χ is a constant having a value of 1.5 and β is given by,

$$\beta = \left(\frac{\Gamma(1+\chi) \times sin\left(\frac{\pi\chi}{2}\right)}{\Gamma\left(\frac{1+\chi}{2}\right) \times \chi \times 2^{\left(\frac{\chi-1}{2}\right)}}\right)^{\frac{1}{\chi}} \tag{29}$$

where, $\Gamma(r) = (r-1)!$.

Step 7) Termination:

The procedure explained above is reiterated until the maximal iteration counts are reached. Using algorithm 1, the pseudo-code of the devised SLnSO algorithm is displayed:

Algorithm 1. Pseudocode of the presented SLnSO algorithm

```
Input : T, c, b, P_qr, Gc, Z_up and Z_low
Output: Z_hc
Begin
    Initiate the population of squirrels using equation (5)
    Determine the fitness of the squirrels with equation (3)
    Sort the squirrels based on their fitness values in increasing order
    Determine the squirrels on hickory, normal and acorn trees.
    While (t < T) do
        For t = 1 : c_1 (c_1 is the count of flying squirrels in acorn tree)
            Update location using equation (6)
        Endfor
        For t = 1 : c_2 (c_2 is the count of flying squirrels in a normal tree which move to the acorn trees)
        If (C_2 > P_qr) then
                Update location using equation (19)
        Else
                Update location to any random location
            Endif
        Endfor
        For t = 1 : c_3 (c_3 is the count of flying squirrels in normal trees which move to the hickory trees)
            Update location using equation (20)
        Endfor
        Compute seasonal constant Z_c using equation (25)
        If (Z_c < Z_min) then
            Update location using equation (27)
        Endif
        Update value of Z_min with equation (26)
    End while
    The optimal solution corresponds to the location of the squirrel at the hickory tree
End
```

Thus, the proposed SLnSO algorithm effectively selects the better features in the input data, where the features are selected and expressed as F and have a dimension of $u \times w$. The integration of the SSA with SLnO effectively enhanced the convergence speed and has attained an enhanced exploration. Then the features are applied to the data augmentation phase.

3-4- Data Augmentation using Bootstrapping

The feature selected F is then subjected to the data augmentation process to increase the sample count in the features to enhance the classifier's effectiveness. Here, bootstrapping [42] is employed for performing data augmentation. Bootstrapping is a re-sampling technique used to create pseudo data for dramatically increasing the training of the DQN. Bootstrap is highly needed to avoid the problem of overfitting in the DQN. The input features are separated into numerous groups, and each group's standard deviation and mean are computed to attain the pseudo data. A normal distribution generates the pseudo-code randomly based on the standard deviation and mean computed. The augmented data thus generated is indicated by D and has a size of $x \times w$, with $x > u$, and this data is fed to the DQN for prediction.

3-5- Performance Prediction to Find At Risk Students using Data Q Network

The augmented data D is fed as the input of the DQN [39], which predicts the at-risk students. The DQN is used to identify the students who are susceptible to failure in the upcoming examinations and who need additional assistance in learning. The DQN utilizes a Q function to estimate the active value function. DQN is highly capable of identifying features in a high-dimensional space, and it offers the benefit of high generalization. Figure 3 displays the structure of the DQN. The input, convolutional, dropout, flattened and dense layers are the numerous layers encompassed by the DQN. DQN is effective in handling instability issues while carrying out reinforcement learning. To tackle these problems, DQN employs a technique called experience replay, by storing the experience of the agent $f_i = (St_i, e_i, n_i, St_{i+1})$ at every step i in the replay memory $E = \{o_1, \dots, o_i\}$. Here, e denotes the reward, St_i and St_{i+1} indicates the state observations. While learning is performed, the experiences $(St_j, e_j, n_j, St_{j+1})$ are acquired from the memory. The loss function given below is employed to update the Q-learning,

$$K_k(\varphi_i) = E\left[\left(n + \lambda \max_{e'} \widehat{U}(St', e'; \varphi_k^-) - U(St, e; \varphi_k)\right)^2\right] \tag{30}$$

where, φ_k and φ_k^- refers to the Q-network parameters in the k^{th} iteration, λ designates the reward factor, employed to estimate the target, $h_j = n + \lambda max_{e'} \ddot{U}\ (St', e';\ \varphi_k^-\ )$ in the k^{th} iteration. The term $n + \lambda \max_{e'} \widehat{U}(St', e'; \varphi_k^-) - U(St, e; \varphi_k)$ denotes the Temporal Difference (TD) error, which affects the stability of convergence. To overcome this, DQN uses a neural fitted Q technique with the parameter φ in the output $n + \lambda max_{e'} \ddot{\bar{U}}\ (St', e';\ \varphi\ )$ considered as fitted $n + \lambda max_{e'} \ddot{U}\ \ (St', e';\ \varphi^-)$ at a specific time. The network parameters φ are updated, once the Q-function is learned. The output thus predicted by the DQN is specified as M, which identifies the at-risk students

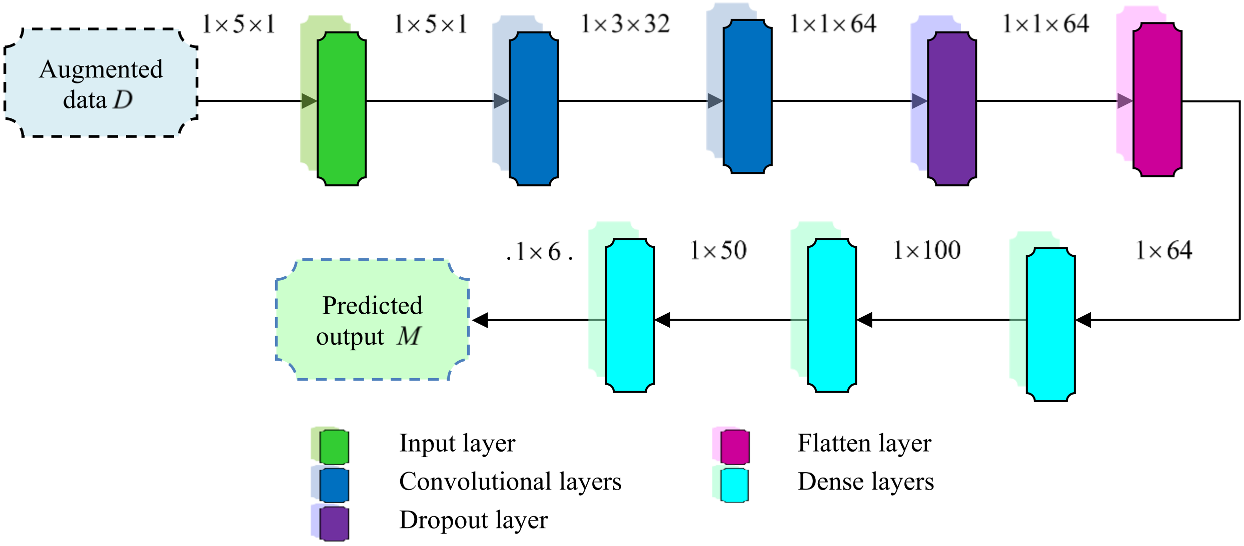


Figure 3. Architecture of DQN

3-5-1- Devised SLnSO Algorithm for Training DQN

The proposed SLnSO algorithm adjusts the weights and bias of DQN. Section 3.3.3 already elaborates SLnSO algorithm. However, the fitness function considered here is calculated based on the Mean Square Error (MSE), given by,

$$MSE = \frac{\sum_{y=1}^{\eta}\left(M_y - M_y^*\right)^2}{\eta} \tag{31}$$

wherein, M and M^* represent the actual and target output of the DQN, and η represents the number of samples.

4- Results and Discussion

From the results, the experimental outputs of the SLnSO-DQN for predicting the at-risk students is discussed. Furthermore, the experimental setup, dataset, evaluation measures, technique, and algorithm assessment are presented in detail.

4-1- Experimental Set-Up

By using Python on a PC with an Intel core i-3 processor, 2GB RAM, and Windows 10 OS, the experimentation of the SLnSO-DQN at-risk predictions are conducted.

4-2- Dataset Description

The dataset utilized for experimentation is the Student performance dataset [40], which contains information regarding students' achievement in two Portuguese schools' secondary education. The information in the dataset was gathered with the help of questionnaires and reports. Further, various attributes, such as school-related, social, demographic features and students' grades, are present in the dataset. Further, it includes the students' performance in two subjects, Portuguese and Mathematics. The data is contained in two files in .csv format and has a total of 30 attributes, and the performance is graded using three grades G1, G2, and G3 (see Table 1).

Table 1. Dataset

Dataset name	Available attributes	Selected attributes
Student Performance Prediction Dataset	School, Sex, Age, Address, Fam size, P status, Medu, Fedu, Mother's job, Father's job, Reason, Guardian, Travel time, Study time, Failures, Schools up, Famsup, Paid, Activities, Nursery, Higher, Internet, Romantic, Fam rel, Free time, Go out, Dalc, Walc, Health and Absences.	School, Sex, Fam size, P status and Guardian

4-3- Evaluation Measures

The metrics, such as MSE, RMSE, and MAE are considered for experimental analysis to determine the efficiency of the developed SLnSO-DQN for at-risk student prediction. The evaluation metrics are shortly detailed below.

MSE: The square of the error between the original and the target outputs is given by MSE and found using Equation 31. MSE is the measure of the closeness of the projected output to the actual output.

RMSE: RMSE is computed by taking the square root of MSE, and is computed by the following expression.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{y=1}^{\eta}\left(M_y - M_y^*\right)^2}{\eta}} \tag{32}$$

MAE: MAE is used to quantify the absolute error between the projected and the targeted output and is calculated by:

$$MSE = \sum_{y=1}^{\eta}\left|M_y - M_y^*\right| \tag{33}$$

4-4- Comparative Techniques

An empirical study is carried out to evaluate the efficiency of the presented SLnSO-DQN at-risk student prediction technique based on several metrics. In addition, the conventional approaches to student performance prediction, such as ML [27], RNN-GRU [2], Deep Learning (DL) [4], CNN [26], Stacked classifier [34], LR-KNN [35], Decision Trees [17], Random Forest [34] and Deep Neural Network (DNN) [43] are considered to examine the efficacy of the developed technique.

4-5- Comparative Assessment

The valuation of the proposed SLnSO-DQN technique for at-risk student prediction is investigated using measures like MSE, RMSE, and MAE, considering k-fold and training data that are described in the section.

- ***Assessment using k-fold***

The evaluation considering the k-fold is displayed in Figure 4. Figure 4-a) depicts the evaluation of MSE by varying the k-fold. With a k-fold of 7, the methods, such as ML, RNN-GRU, Stacked classifier, DL, LR-KNN, Decision Trees, CNN, Random Forest, DNN, and SLnSO-DQN attained an MSE of 0.954, 0.714, 0.688, 0.446, 0.398, 0.288, 0.260, 0.246, 0.240, and 0.238. Figure 4-b) exhibits the RMSE-based assessment. When a k-fold value of 6 is taken, the RMSE value calculated is 0.978 for ML, 0.841 for RNN-GRU, 0.858 for Stacked classifier, 0.665 for DL, 0.639 for LR-KNN, 0.554 for Decision Trees, 0.526 for CNN, 0.498 for Random Forest, 0.492 for DNN, and 0.486 for the SLnSO-DQN. In Figure 4-c), the analysis considering MAE is presented. The value of MAE measured by various students' performance prediction approaches, like ML is 0.869, RNN-GRU is 0.713, Stacked classifier is 0.537, DL is 0.475, LR-KNN is 0.426, Decision Trees is 0.418, CNN is 0.409, Random Forest is 0.326, DNN is 0.316, and SLnSO-DQN is 0.285 for K-Fold=8. From this evaluation, it is clear that the proposed SLnSO-DQN model consistently achieves the lowest error rates for MSE, RMSE, and MAE and outperforms ML, RNN-GRU, Stacked classifier, DL, LR-KNN, Decision Trees, CNN, Random Forest, and DNN. In the proposed model, the SLnSO is developed to train the DQN to achieve these better results.

- ***Assessment based on training data***

In Figure 5, the comparative assessment of different training data values is depicted. The analysis of MSE is displayed using Figure 5-a). The value of MSE calculated for 80% training data is 0.924 for ML, 0.731 for RNN-GRU, 0.548 for Stacked classifier, 0.457 for DL, 0.327 for LR-KNN, 0.300 for Decision Trees, 0.291 for CNN, 0.259 for Random Forest, 0.254 for DNN, and 0.244 for the SLnSO-DQN. Figure 5-b) portrays the evaluation by changing the training data. The conventional techniques, such as ML, RNN-GRU, Stacked classifier, DL, LR-KNN, Decision Trees, CNN, Random Forest, DNN estimated RMSE of 0.957, 0.854, 0.719, 0.675, 0.557, 0.526, 0.511, 0.506, and 0.500 with 90% training data, whereas the SLnSO-DQN gauged a lower value of RMSE at 0.493. In Figure 5-c), the analysis of MAE is displayed. The various schemes, like ML, RNN-GRU, Stacked classifier, DL, LR-KNN, Decision Trees, CNN, Random Forest, DNN, and SLnSO-DQN, achieved MAE of 0.876, 0.767, 0.648, 0.511, 0.458, 0.439, 0.388, 0.369, 0.348, and 0.307, correspondingly, for 70% of training data. The SLnSO-DQN effectively captures hidden patterns in student performance data, which leads to consistently lower MSE, RMSE, and MAE across different training sizes.

Also, the SLnSO-DQN maintains stable performance even with changes in training data proportions, which makes it a reliable choice for early student dropout prediction and academic success forecasting.

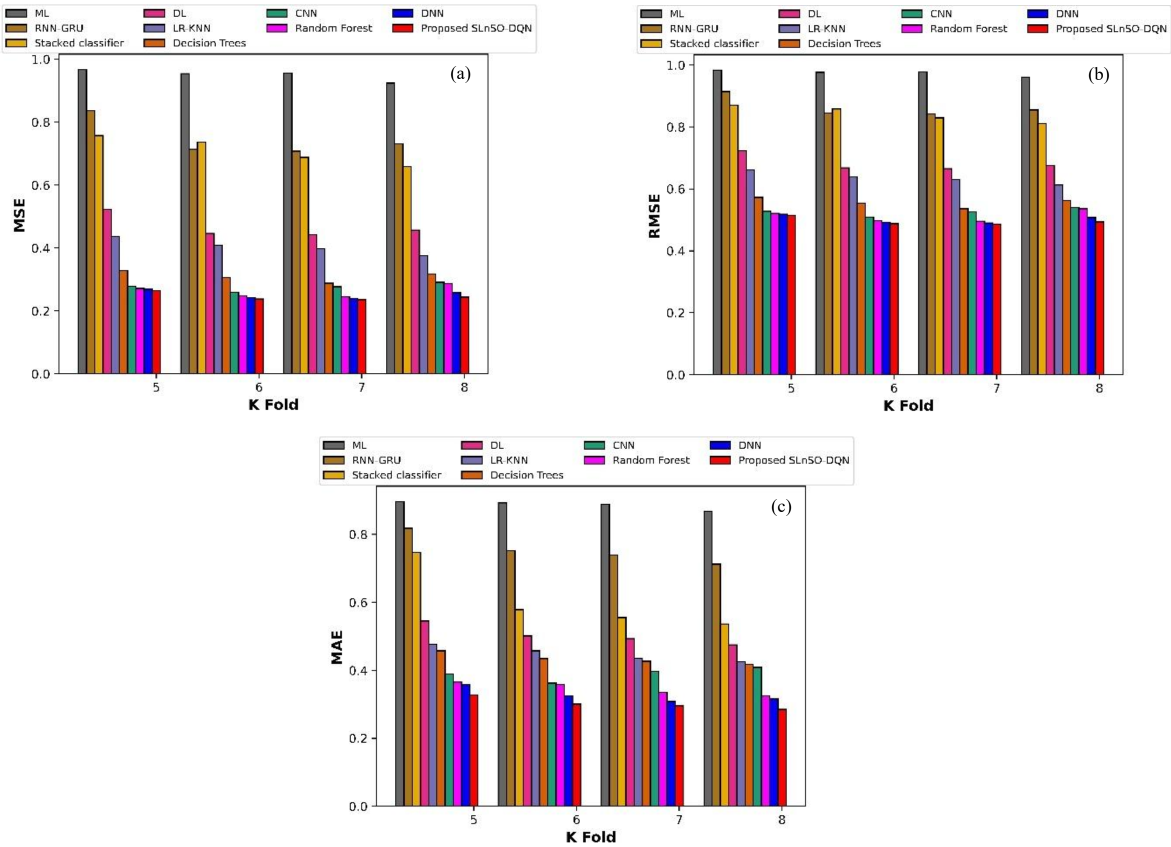


Figure 4. **Analysis of the SLnSO-DQN using k-fold based on a) MSE, b) RMSE, and c) MAE**

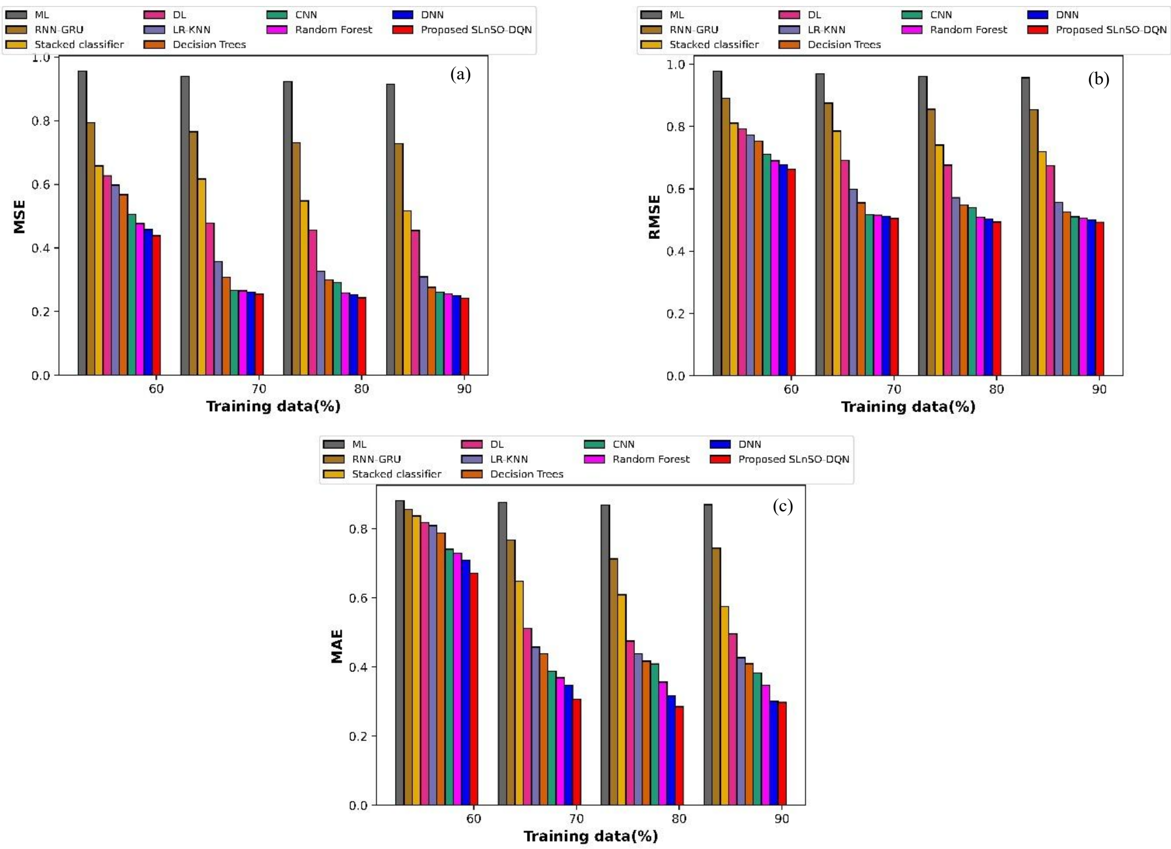
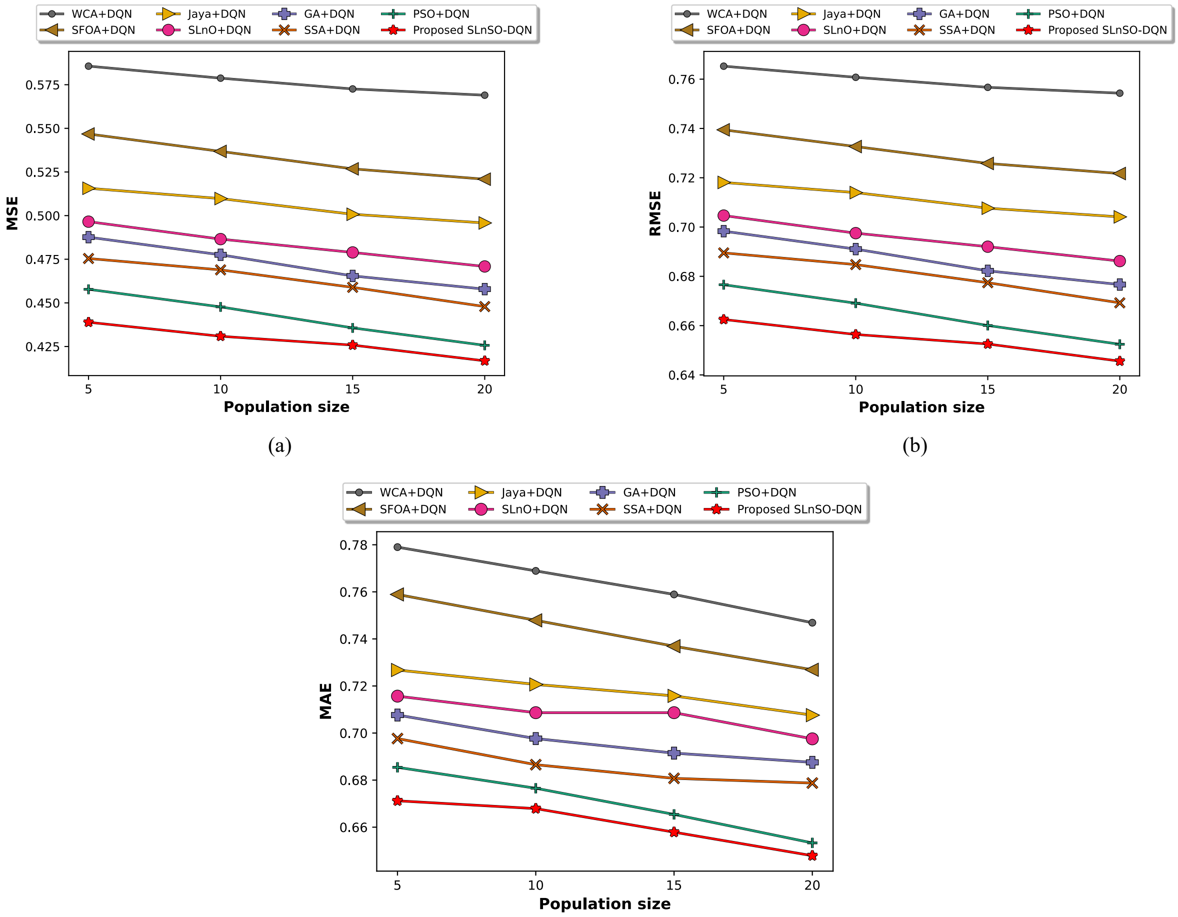


Figure 5. **Evaluation of the SLnSO-DQN concerning training data based on a) MSE, b) RMSE, and c) MAE**

4-6- Algorithmic Assessment

The performance of the SLnSO is examined by comparing it with various algorithms, like Water Cycle Algorithm [44], SailFish Optimizer Algorithm (SFOA) [45], Jaya [46], SLnO [38], Genetic Algorithm (GA) [47], SSA [37], and Particle Swarm Optimization (PSO) [48], considering various metrics by varying the population size. In Figure 6-a), the analysis is portrayed based on MSE. The MSE values computed by the WCA+DQN, SFOA+DQN, Jaya+DQN, SLnO+DQN, GA+DQN, SSA+DQN, PSO+DQN and SLnSO+DQN is 0.569, 0.521, 0.496, 0.471, 0.458, 0.448, 0.426, and 0.417, respectively with a population size of 20. Figure 6-b) depicts the assessment of RMSE. With a population size of 15, the RMSE value attained is 0.757 for WCA+DQN, 0.726 for SFOA+DQN, 0.708 for Jaya+DQN, 0.692 for SLnO+DQN, 0.682 for GA+DQN, 0.677 for SSA+DQN, 0.660 for PSO+DQN and 0.653 for SLnSO+DQN. Figure 6-c) displays the assessment of MAE. The value of MAE achieved by WCA+DQN is 0.769, SFOA+DQN is 0.748, Jaya+DQN is 0.721, SLnO+DQN is 0.709, GA+DQN is 0.698, SSA+DQN is 0.687, PSO+DQN is 0.677, and the SLnSO+DQN is 0.668 with population size 10. Here, the SLnSO combines SLnO and SSA, which leads to more effective hyperparameter tuning and reduced error values across all metrics. Also, the hybrid SLnSO+DQN model adapts to different population sizes, which maintains stability in error minimization across MSE, RMSE, and MAE.



(a) (b)

(c)

Figure 6. Analysis of the SLnSO based on a) MSE, b) RMSE, and c) MAE

4-7- Comparative Discussion

This section explains the comparative discussion of the SLnSO-DQN for at-risk student prediction. The SLnSO-DQN is scrutinized for its efficacy in view of various metrics that include MSE, RMSE and MAE, based on different training data values and k-fold, which is displayed in Table 2. The values listed in the table correlate to a k-fold of 9, and the training data is 90%. This is detected from the table that the proposed SLnSO-DQN achieved a minimal value of MSE, RMSE and MAE of 0.265, 0.514, and 0.327, respectively. The low value of MSE is accounted for by the utilization of the SLnSO for feature selection, resulting in reduced RMSE. The usage of DQN for predicting at-risk students effectively minimized the MAE.

Table 2. Comparative discussion of the SLnSO-DQN for at-risk student prediction

Variations	*Metrics*	*ML*	*RNN-GRU*	*Stacked classifier*	*DL*	*LR-KNN*	*Decision Trees*	*CNN*	*Random Forest*	*DNN*	*SLnSO-DQN*
K-fold	MSE	0.967	0.836	0.757	0.523	0.437	0.328	0.279	0.272	0.269	**0.265**
	MAE	0.896	0.818	0.747	0.545	0.477	0.458	0.390	0.366	0.358	**0.327**
	RMSE	0.983	0.915	0.870	0.723	0.661	0.573	0.528	0.521	0.518	**0.514**
Training data	MSE	0.956	0.794	0.658	0.627	0.598	0.568	0.506	0.477	0.458	0.439
	MAE	0.881	0.855	0.837	0.818	0.809	0.788	0.740	0.729	0.709	0.671
	RMSE	0.978	0.891	0.811	0.792	0.773	0.754	0.711	0.690	0.677	0.662

In Table 3, the comparison of the SLnSO is displayed. The algorithmic assessment is carried out using various metrics, like MSE, RMSE and MAE, considering different population sizes.

Table 3. Comparative discussion of the SLnSO

Metrics	*WCA+DQN*	*SFOA+DQN*	*Jaya+DQN*	*SLnO+DQN*	*GA+DQN*	*SSA+DQN*	*PSO+DQN*	*SLnSO+DQN*
MSE	0.586	0.547	0.516	0.497	0.488	0.475	0.458	**0.439**
MAE	0.779	0.759	0.727	0.716	0.708	0.698	0.685	**0.671**
RMSE	0.765	0.739	0.718	0.705	0.698	0.690	0.677	**0.662**

5- Conclusion

The increasing trend of online education has made it crucial to ensure students successfully complete their courses on time. Early finding of at-risk students plays a crucial role in offering timely interventions and support. This paper proposes an effective technique for predicting at-risk students, named SLnSO-DQN. The at-risk students are identified by considering various attributes such as demographic, social, and school-related features along with the student grade. Here, the SLnSO is utilized for selecting the salient features in the input data based on the DL distance as fitness. Further, the at-risk students are predicted by using the DQN, which is trained using the SLnSO. The effectiveness of the SLnSO-DQN shows a minimum value of MSE at 0.265, RMSE at 0.514, and MAE at 0.327. The proposed model is used to help educational institutions find at-risk students early, which enables timely interventions such as personalized academic support, counselling, and mentoring programs. Also, it helps educators to continuously track students' progress, and learning difficulties detection, and modify the teaching methodologies accordingly. However, the required computational requirements of the model are increased when using large-scale data. It limits the effectiveness of the proposed model in real-time applications. Also, in this research, the evaluation is conducted using a single dataset, and the applicability of the model in different institutions with varying educational structures and grading systems is not considered. These limitations will be considered in the further extension of this work. To improve the overall performance of the model, the evaluation will be conducted across multiple educational institutions. Additionally, other augmentation methods, such as the Synthetic Minority Over-sampling Technique (SMOTE) and generative Adversarial Network (GAN)-based approaches, are combined with bootstrapping to better maintain real correlations in the further extension of this work.

6- Declarations

6-1- Author Contributions

Conceptualization, P.V., J.M., and B.K.; methodology, P.V., J.M., and B.K.; formal analysis, P.V., J.M., H.S., and R.R.; investigation, P.V., J.M., H.S., and R.R.; data curation, P.V., H.S., and B.K.; writing—original draft preparation, P.V., B.K., and R.R.; writing—review and editing, P.V., B.K., and R.R. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data that support the findings of this study are openly available in Student performance dataset at https://archive.ics.uci.edu/ml/datasets/student+performance#.

6-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Kaur, G., & Singh, W. (2016). Prediction of student performance using weka tool. An International Journal of Engineering Sciences, 17, 8-16.

[2] He, Y., Chen, R., Li, X., Hao, C., Liu, S., Zhang, G., & Jiang, B. (2020). Online at-risk student identification using RNN-GRU joint neural networks. Information (Switzerland), 11(10), 1–11. doi:10.3390/info11100474.

[3] Mahajan, G., & Saini, B. (2020). Educational Data Mining: A state-of-the-art survey on tools and techniques used in EDM. International Journal of Computer Applications & Information Technology, 12(1), 310-316.

[4] Dien, T. T., Luu, S. H., Thanh-Hai, N., & Thai-Nghe, N. (2020). Deep learning with data transformation and factor analysis for student performance prediction. International Journal of Advanced Computer Science and Applications, 11(8), 711–721. doi:10.14569/IJACSA.2020.0110886.

[5] Chen, Y., & Zhai, L. (2023). A comparative study on student performance prediction using machine learning. Education and Information Technologies, 28(9), 12039–12057. doi:10.1007/s10639-023-11672-1.

[6] Su, Y., Liu, Q., Liu, Q., Huang, Z., Yin, Y., Chen, E., Ding, C., Wei, S., & Hu, G. (2018). Exercise-enhanced sequential modeling for student performance prediction. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 32(1), 2435–2443. doi:10.1609/aaai.v32i1.11864.

[7] Alshammari, A. (2024). Using analytics to predict students' interactions with learning management systems in online courses. Education and Information Technologies, 29(15), 20587–20612. doi:10.1007/s10639-024-12709-9.

[8] Thai-Nghe, N., Horváth, T., & Schmidt-Thieme, L. (2011). Factorization Models for Forecasting Student Performance. 4th international conference on educational data mining, 6-8 July, 2011, Eindhoven, Netherlands.

[9] Liu, D., Zhang, Y., Zhang, J., Li, Q., Zhang, C., & Yin, Y. (2020). Multiple features fusion attention mechanism enhanced deep knowledge tracing for student performance prediction. IEEE Access, 8, 194894–194903. doi:10.1109/ACCESS.2020.3033200.

[10] Mat, U. B., Buniyamin, N., Arsad, P. M., & Kassim, R. (2013). An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention. 2013 IEEE 5th Conference on Engineering Education (ICEED), 126–130. doi:10.1109/iceed.2013.6908316.

[11] Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students' performance by means of machine learning: A review. Applied Sciences (Switzerland), 10(3), 1042. doi:10.3390/app10031042.

[12] Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science, 72, 414–422. doi:10.1016/j.procs.2015.12.157.

[13] Chicon, P. M. M., Paschoal, L. N., Sawicki, S., Roos-Frantz, F., & Frantz, R. Z. (2025). A Predictive Model for the Early Identification of Student Dropout Using Data Classification, Clustering, and Association Methods. Revista Iberoamericana de Tecnologias Del Aprendizaje, 20, 12–21. doi:10.1109/RITA.2025.3528369.

[14] Kemper, L., Vorhoff, G., & Wigger, B. U. (2020). Predicting student dropout: A machine learning approach. European Journal of Higher Education, 10(1), 28–47. doi:10.1080/21568235.2020.1718520.

[15] Kim, B. H., Vizitei, E., & Ganapathi, V. (2018). GritNet: Student performance prediction with deep learning. arXiv (Preprint), arXiv.1804.07405. doi:10.48550/arXiv.1804.07405.

[16] Dixit, P., Nagar, H., & Dixit, S. (2022). Student performance prediction using case based reasoning knowledge base system (CBR-KBS) based data mining. International Journal of Information and Education Technology, 12(1), 30–35. doi:10.18178/ijiet.2022.12.1.1583.

[17] Alsariera, Y. A., Baashar, Y., Alkawsi, G., Mustafa, A., Alkahtani, A. A., & Ali, N. (2022). Assessment and Evaluation of Different Machine Learning Algorithms for Predicting Student Performance. Computational Intelligence and Neuroscience, 2022, 1–11. doi:10.1155/2022/4151487.

[18] Kumar Veerasamy, A., D'Souza, D., Apiola, M.-V., Laakso, M.-J., & Salakoski, T. (2020). Using early assessment performance as early warning signs to identify at-risk students in programming courses. 2020 IEEE Frontiers in Education Conference (FIE), 1–9. doi:10.1109/fie44824.2020.9274277.

[19] Zerkouk, M., Mihoubi, M., Chikhaoui, B., & Wang, S. (2024). A machine learning based model for student's dropout prediction in online training. Education and Information Technologies, 29(12), 15793–15812. doi:10.1007/s10639-024-12500-w.

[20] Albreiki, B., Habuza, T., Palakkal, N., & Zaki, N. (2024). Clustering-based knowledge graphs and entity-relation representation improves the detection of at risk students. Education and Information Technologies, 29(6), 6791–6820. doi:10.1007/s10639-023-11938-8.

[21] Pei, B., & Xing, W. (2022). An Interpretable Pipeline for Identifying At-Risk Students. Journal of Educational Computing Research, 60(2), 380–405. doi:10.1177/07356331211038168.

[22] Wang, X., Guo, B., & Shen, Y. (2022). Predicting the At-Risk Online Students Based on the Click Data Distribution Characteristics. Scientific Programming, 2022, 1–12. doi:10.1155/2022/9938260.

[23] Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learning Environments, 9(1), 1–19. doi:10.1186/s40561-022-00192-z.

[24] Parhizkar, A., Tejeddin, G., & Khatibi, T. (2023). Student performance prediction using datamining classification algorithms: Evaluating generalizability of models from geographical aspect. Education and Information Technologies, 28(11), 14167–14185. doi:10.1007/s10639-022-11560-0.

[25] Susheelamma, K. H., & Ravikumar, K. M. (2019). Student risk identification learning model using machine learning approach. International Journal of Electrical and Computer Engineering, 9(5), 3872–3879. doi:10.11591/ijece.v9i5.pp3872-3879.

[26] Yang, Z., Yang, J., Rice, K., Hung, J. L., & Du, X. (2020). Using Convolutional Neural Network to Recognize Learning Images for Early Warning of At-Risk Students. IEEE Transactions on Learning Technologies, 13(3), 617–630. doi:10.1109/TLT.2020.2988253.

[27] Azcona, D., Hsiao, I. H., & Smeaton, A. F. (2019). Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints. User Modeling and User-Adapted Interaction, 29(4), 759–788. doi:10.1007/s11257-019-09234-7.

[28] Son, L. H., & Fujita, H. (2019). Neural-fuzzy with representative sets for prediction of student performance. Applied Intelligence, 49(1), 172–187. doi:10.1007/s10489-018-1262-7.

[29] Kustitskaya, T. A., Kytmanov, A. A., & Noskov, M. V. (2020). Student-at-risk detection by current learning performance indicators using Bayesian networks. arXiv (Preprint), arXiv.2004.09774. doi:10.48550/arXiv.2004.09774.

[30] Leelaluk, S., Tang, C., Minematsu, T., Taniguchi, Y., Okubo, F., Yamashita, T., & Shimada, A. (2024). Attention-Based Artificial Neural Network for Student Performance Prediction Based on Learning Activities. IEEE Access, 12, 100659–100675. doi:10.1109/access.2024.3429554.

[31] Mehdi, R., & Nachouki, M. (2023). A neuro-fuzzy model for predicting and analyzing student graduation performance in computing programs. Education and Information Technologies, 28(3), 2455–2484. doi:10.1007/s10639-022-11205-2.

[32] Ouyang, F., Wu, M., Zheng, L., Zhang, L., & Jiao, P. (2023). Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course. International Journal of Educational Technology in Higher Education, 20(1), 4. doi:10.1186/s41239-022-00372-4.

[33] Kusumawardani, S. S., & Alfarozi, S. A. I. (2023). Transformer Encoder Model for Sequential Prediction of Student Performance Based on Their Log Activities. IEEE Access, 11, 18960–18971. doi:10.1109/ACCESS.2023.3246122.

[34] Noviandy, T. R., Zahriah, Z., Yandri, E., Jalil, Z., Yusuf, M., Mohamed Yusof, N. I. S., Lala, A., & Idroes, R. (2024). Machine Learning for Early Detection of Dropout Risks and Academic Excellence: A Stacked Classifier Approach. Journal of Educational Management and Learning, 2(1), 28–34. doi:10.60084/jeml.v2i1.191.

[35] Awedh, M. H., & Mueen, A. (2025). Early Identification of Vulnerable Students with Machine Learning Algorithms. WSEAS Transactions on Information Science and Applications, 22, 166–188. doi:10.37394/23209.2025.22.16.

[36] Othman, S. A., & Ali, H. T. M. (2021). Improvement of the Nonparametric Estimation of Functional Stationary Time Series Using Yeo-Johnson Transformation with Application to Temperature Curves. Advances in Mathematical Physics, 2021, 1–6. doi:10.1155/2021/6676400.

[37] Jain, M., Singh, V., & Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm and Evolutionary Computation, 44, 148–175. doi:10.1016/j.swevo.2018.02.013.

[38] Masadeh, R., Mahafzah, B. A., & Sharieh, A. (2019). Sea Lion Optimization algorithm. International Journal of Advanced Computer Science and Applications, 10(5), 388–395. doi:10.14569/ijacsa.2019.0100548.

[39] Sasaki, H., Horiuchi, T., & Kato, S. (2017). A study on vision-based mobile robot learning by deep Q-network. 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2017, 2017-November, 799–804. doi:10.23919/SICE.2017.8105597.

[40] UCI Machine Learning Repository (2025). Student performance dataset. UCI Machine Learning Repository, Noida, India. Available online: https://archive.ics.uci.edu/ml/datasets/student+performance# (accessed on July 2025).

[41] Zhao, C., & Sahni, S. (2019). String correction using the Damerau-Levenshtein distance. BMC Bioinformatics, 20(S11), 277. doi:10.1186/s12859-019-2819-0.

[42] Efron, B., & Tibshirani, R. J. (1993). Introduction. An Introduction to the Bootstrap, 1–9, Chapman and Hall/CRC, New York, United States. doi:10.1007/978-1-4899-4541-9_1.

[43] Nabil, A., Seyam, M., & Abou-Elfetouh, A. (2021). Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks. IEEE Access, 9, 140731–140746. doi:10.1109/ACCESS.2021.3119596.

[44] Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers and Structures, 110–111, 151–166. doi:10.1016/j.compstruc.2012.07.010.

[45] Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Engineering Applications of Artificial Intelligence, 80, 20–34. doi:10.1016/j.engappai.2019.01.001.

[46] Venkata Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, 7(1), 19–34. doi:10.5267/j.ijiec.2015.8.004.

[47] Mirjalili, S. (2019). Genetic Algorithm. Evolutionary Algorithms and Neural Networks. Studies in Computational Intelligence, vol 780. Springer, Cham, Switzerland. doi:10.1007/978-3-319-93025-1_4.

[48] Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. Expert Systems with Applications, 38(8), 10161–10169. doi:10.1016/j.eswa.2011.02.075.