Revolutionizing Education: Cutting-Edge Predictive Models for Student Success

Moyan Li[1], Suyawen[2]*
School of Culture and Tourism, Shantou Polytechnic, Shantou 515000, Guangdong, China[1]
Normal College, Jimei University, Xiamen 361000, Fujian, China[2]

***Abstract*—Student performance prediction systems are crucial for improving educational outcomes in various institutions, including universities, schools, and training centers. These systems gather data from diverse sources such as examination centers, registration departments, virtual courses, and e-learning platforms. Analyzing educational data is challenging due to its vast and varied nature, and to address this, machine learning techniques are employed. Dimensionality reduction, enabled by machine learning algorithms, simplifies complex datasets, making them more manageable for analysis. In this study, the Support Vector Classification (SVC) model is used for student performance prediction. SVC is a powerful machine-learning approach for classification tasks. To further enhance the model's efficiency and accuracy, two optimization algorithms, the Sea Horse Optimization (SHO) and the Adaptive Opposition Slime Mould Algorithm (AOSMA), are integrated. Machine learning (ML) reduces complexity through techniques like feature selection and dimensionality reduction, improving the effectiveness of student performance prediction systems and enabling data-informed decisions for educators and institutions. The combination of SVC with these innovative optimization strategies highlights the study's commitment to leveraging the latest advancements in ML and bio−inspired algorithms for more precise and robust student performance predictions, ultimately enhancing educational outcomes. Based on the obtained outcomes, it reveals that the SVSH model registered the best performance in predicting and categorizing the student performance with Accuracy=92.4%, Precision=93%, Recall=92%, and F1_Score=92%. Implementing SHO and AOSMA optimizers to the SVC model resulted in improvement of Accuracy evaluator outputs by 2.12% and 0.89%, respectively.**

Keywords—Student performance;* Support Vector Classification*; sea horse optimization; adaptive opposition slime mould algorithm

I. Introduction

Academic information systems, e-learning, and admissions systems are all contributing to the growth of educational data [1]. But since it's so large and intricate, a lot of this data gets wasted. Predicting student achievement requires careful examination of these data [2]. KDD, or knowledge discovery in databases, is another name for data mining (DM) has been successfully applied in various domains, including education, leading to the field of Educational Data Mining (EDM) [3], [4].

Predicting student performance is a crucial endeavor in education, primarily employing EDM [5] to forecast outcomes like passing, failing, and grades. Creating an early warning system to save expenses, save time, and maximize resources is a major emphasis in this field. By enabling educators to modify their teaching strategies and provide more assistance to students who need it, improved educational procedures may raise student achievement [6]. Students are better able to comprehend their probable course performance and make the necessary decisions thanks to these projections. Increasing student retention is one of the institution's long-term objectives as it improves graduates' reputations, rankings, and employment chances [7]. Educational institutions employ DM, often referred to as EDM, to analyze accessible data [8]. Machine learning (ML) algorithms provide essential tools for knowledge discovery [9]. Predicting performance accurately helps identify difficult pupils early on. By analyzing educational data, EDM supports institutions in making improvements and creating new teaching strategies. [10]. Predicting academic success, however, is difficult since there are many different elements that might influence it [11]. Technological developments have made it possible to create efficient ML techniques. New studies demonstrate how effective ML methods are in enhancing instruction [12].

II. Related Works

Carlos et al. [13] used ML to create a student failure forecast model, achieving the highest accuracy (92.7%) with the ICRM classifier. However, they did not test the model on different educational levels due to varying student characteristics. Dorina et al. [14] created a classifier-based forecasting model for student performance. While other models fared better in identifying failure students, the MLP model had the best accuracy (73.59%) in identifying successful students. Class balance and high-dimensional data presented challenges for the model. Osmanbegovic and Suljic [15] created a model that accounts for data dimensionality and forecasts academic achievement in students. After testing many classifiers, Naïve Bayes achieved the maximum accuracy of 76.65% ; nevertheless, the model did not address the problem of class imbalance. In addressing course dropouts, an EDM challenge [16] employed four data mining methods with various attribute combinations. With the use of certain predictors, the support vector machine model produced the best accurate categorization. However, because student knowledge may have grown throughout the course, it was limited to incorporating earned marks from required courses. Ajay et al. [17] researched predicting student performance, introducing the "CAT" social factor. This factor classifies Indians based on social status, which influences education. They employed four classifiers

(OneR, MLP, J48, and IB1) on the dataset, with the $IB1$ model achieving the highest accuracy at 82%.

Ramanathan et al. [18] aimed to enhance the $ID3$ model for forecasting student academic performance. The ID3 model's weakness was inefficiently selecting attributes with numerous values as nodes, resulting in suboptimal trees. The proposed model addressed this issue and produced two output classes (Pass and Fail). Upon testing many classifiers, including J48, wID3, and Naïve Bayes, the wID3 classifier demonstrated an impressive 93% accuracy. Dech Thammasiri et al. [19] introduced a model to predict poor academic performance among freshmen. They utilized four classification methods and three balancing techniques to address class imbalance. The most accurate result, with 90.24% overall accuracy, was achieved by combining the support vector machine with SMOTE.

The research proposed a prediction approach for online student learning performance utilizing learning portfolio data [20]. The findings showed that time-dependent variable-incorporating approaches were more accurate than those that did not. It is important to remember, nonetheless, that the model was not evaluated in an offline mode, when the introduction of time-dependent characteristics would have led to a drop in performance. Contrary to previous assumptions, Natek and Zwilling [21] emphasized data mining's suitability for tiny datasets. It demonstrated a model for predicting student achievement using three decision tree techniques and a small dataset, with Reptree obtaining an accuracy rate of more than 90%.

However, the model did not handle issues with class balance or large data dimensionality. Marbouti et al. [22] introduced an ensemble model for identifying underperforming students, comprising classifiers like NB, SVM, and KNN. The dataset featured a crucial attribute: standard-based grading assessment alongside the usual score-based grading. When compared to six individual classifiers, the ensemble model achieved the highest accuracy at 85%. To address multiclass classification issues in student performance prediction, a multi-level model was proposed in a study [23]. Enhancing both the overall model accuracy and the accuracy of each classifier separately was the aim. The model has two stages: J48 was chosen for the subsequent level after resampling and four classifiers were used in the first level. After removing outliers, resampling with J48 at the second level produced predictions for each class that were above 90% accurate overall. Costa et al. [24] introduced a model for early student failure diagnosis that evaluates preprocessing and data mining strategies. ANNs, decision trees, support vector machines, and naïve Bayes were among the models and approaches used. Support vector machines fared better than the others, according to the findings. Although information was gathered from two different sources, the model did not take the decrease in categorization mistakes into account.

This research is paramount in its aim to develop a sophisticated ML model for predicting student performance, leveraging data from reliable sources. The cornerstone of this study is the implementation of the Support Vector Classification (SVC) technique, chosen for its effectiveness in handling the inherent complexities of high-dimensional datasets in the educational domain. The decision to focus on student performance prediction is underscored by the critical role it plays in shaping educational outcomes. What sets this study apart is the innovative integration of two optimization algorithms, namely the Sea Horse Optimization (SHO) and the Adaptive Opposition Slime Mould Algorithm (AOSMA), seamlessly woven into the fabric of the SVC model. This unique combination of techniques represents a novel approach, seeking not only to predict student performance but to elevate the precision and accuracy of the predictive model. The integration of SHO and AOSMA introduces a layer of sophistication, bringing forth the potential to enhance the model's predictive capabilities. These optimization algorithms are strategically applied, each contributing its unique strengths to the overall optimization process. The SHO algorithm, inspired by the efficient and adaptive nature of sea-horses, aims to refine the predictive model by iteratively fine-tuning parameters.

On the other hand, the AOSMA, drawing inspiration from the efficient behaviors of slime molds, further contributes by guiding the model toward optimal solutions. SVC emerges as a fitting choice for predicting student performance due to its ability to discern non-linear relationships within intricate datasets. It operates by identifying decision boundaries that maximize the separation between different performance classes, enabling the classification of students into distinct categories such as success or failure. The intricate nature of educational data demands a tool that can navigate through complexities, and SVC proves to be a valuable asset in this regard. The ultimate goal of this study is not only to predict student performance accurately but also to contribute to the broader landscape of educational decision-making. The integration of cutting-edge optimization algorithms with a powerful machine learning technique like SVC positions this research at the forefront of innovation in educational data analysis. By seamlessly blending theory and practice, this study offers a glimpse into the potential advancements that can be made in refining and improving predictive models for student performance in educational settings. Related works is given in Section II. Section III delves into research methodology. An elaborate explanation of the data and an assessment of the models based on metrics will be provided. In Section IV, the results derived from the training and testing phases will be scrutinized, and subsequently, the performance of the models based on classification will be reported. Finally, in Section V, conclusions regarding the study in question and the overall performance of the models will be presented.

III. Research Methodology

A. Data Processing

Creating a reliable approach for precisely evaluating students' academic performance and the several contextual elements that affect it is the main goal of this project. This can only be accomplished by doing necessary preprocessing on the original dataset. First, textual input must be transformed into numerical values. This is a necessary precondition for doing machine learning tasks. This translation enables the use of sophisticated statistical methods and aids efficient data processing. 649 datasets are included in the dataset, which includes a wide range of characteristics that may have an impact on students' academic performance. These variables include school, sex, age, residence in an urban or rural area ($address$), parental cohabitation status ($Pstatus$), family size ($famsize$), parental education and occupations ($Medu, Fedu, Mjob,$ and $Fjob$), school choice motivation (reason), guardian, travel time from home to school, study time each week, past class failures ($failures$), participation in supplemental education ($schoolsup$), family educational support ($famsup$), extracurricular activities, nursery school attendance, aspiration for higher education, internet access, romantic relationships, family relationship quality, free time, socializing frequency, weekday ($Dalc$) and weekend ($Walc$) alcohol consumption, and student absences. This study's main objective is to forecast and categorize students' academic achievement based on the variable $G3$, which represents final grades from school reports that range from zero (lowest grade) to twenty (highest grade). Four separate levels are assigned to these grades: A more detailed evaluation of student success is made possible by the classifications of Poor $(0-12)$, Acceptable $(12-14)$, Good $(14-16)$, and Excellent $(16-20)$. In the end, this method aims to improve educational practices and policy formation by offering a thorough framework for comprehending and measuring academic achievement within a variety of contextual elements.

Fig. 1 presents a correlation matrix encompassing input and output variables in this study. Study time positively impacted academic performance, while previous failures had a negative effect. Internet access and aspirations for higher education had positive influences, contrasting alcohol consumption's negative impact. Parental education, particularly mothers', positively affected grades. Daily/weekly alcohol consumption, past failures, and student age influenced school grades.

In conclusion, the matrix emphasizes how crucial study time and parental education are to scholastic achievement. The dataset was obtained from two secondary schools for Mathematics subject [25]. It comprised 32 input features, including demographic information, social features, and grades, along with a single output denoted as the final grade (G3). Datasets were amalgamated to facilitate a feature selection method in this study. The dataset underwent simplification through the normalization of input features within the range of [0,1].

In the data preprocessing phase, the inherent complexity of educational data was addressed through a robust pipeline. The process included identifying and handling missing or erroneous data points to ensure dataset integrity. Additionally, numerical features were standardized to a common scale to prevent bias arising from varying magnitudes. Categorical variables were encoded to facilitate machine learning algorithms in interpreting and learning from the data.

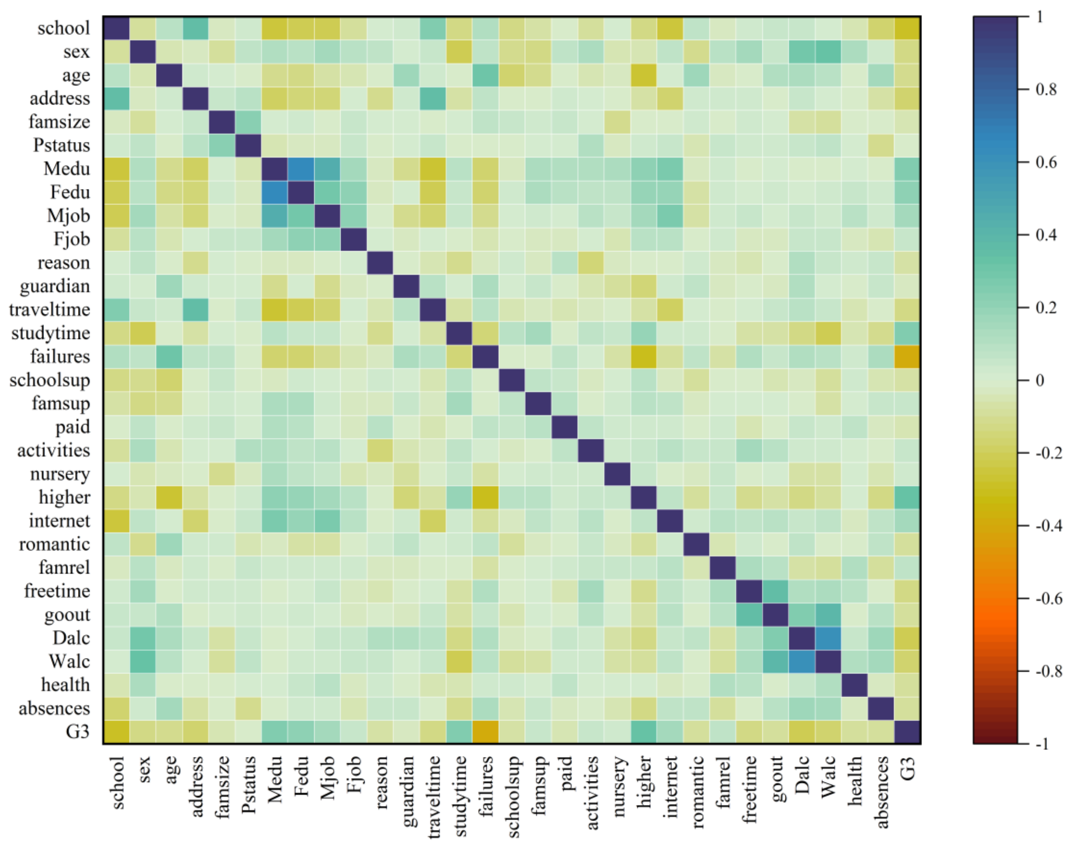


Fig. 1. Correlation matrix for the input and output variables.

B. Evaluation of Models' Applicability

Accuracy is a widely used indicator to evaluate a model's overall performance in classification challenges. True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) are its four essential building blocks. Correct forecasts are represented by TP, accurate negative predictions by TN, inaccurate positive predictions by FP, and incorrect negative predictions by FN.

Accuracy, however, has limits when dealing with uneven data since it favors the majority class and offers no new information. Three further assessment metrics Precision, Recall, and F1-Score are used to solve this.

1) Recall: This metric evaluates a model's ability to identify all relevant instances within a specific class correctly. It is crucial for reducing FN, instances that should be identified but are missed.

2) Precision: Precision assesses the accuracy of positive predictions made by the model, reducing False Positives, which are instances predicted as positive but do not belong to the class.

3) F1-Score: A fair evaluation of the model's performance is provided by the F1-Score, which combines Precision and Recall. When considering both minority and majority classes in unbalanced data sets, it is invaluable.

Together, these metrics which are described by mathematical formulas (Eq. (1) through Eq. (4)) offer a more thorough knowledge of the efficacy of a categorization model. They are especially helpful in addressing class disparities that may skew how accuracy is interpreted. Researchers and data analysts may enhance model performance by using these indicators to make better-informed judgments and modifications, especially in challenging imbalanced data situations.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = TPR = \frac{TP}{P} = \frac{TP}{TP+FN} \tag{3}$$

$$F1\_score = \frac{2 \times Recall \times Precision}{Recall+Precision} \tag{4}$$

C. Support Vector Classification (SVC)

Support Vector Classification is an algorithm rooted in the structured principle of minimizing risk within the framework of support vector machines [26]. Non-linear transformations are applied to the independent variables, projecting them into a high-dimensional space. In this space, an optimal hyperplane is constructed to separate both classes. The primary goal of this hyperplane is to minimize classification errors while simultaneously maximizing the margins, which represent the total distance from the hyperplane to the closest training samples of each class [27].

The primary model is subsequently shown in Eq. (5) to Eq. (7) [28].

$$min_{w,b,\in} \frac{\|W\|^2}{2} + C_{svc} \sum_{i=1}^{N} \in_i \tag{5}$$

$$y_i(w^T.\emptyset(x_i) + b) \geq 1 - \in_i \qquad i = 1,\ldots,N \tag{6}$$

$$\in_i \geq 0 \qquad i = 1,\ldots,N \tag{7}$$

The function $\emptyset(x_i)$ is a non-linear transformation that takes each observation, defined by its explanatory variables x_i, and projects it into a higher-dimensional space.

C_{svc} shows a regularization parameter

w represents the weight vector related to the explanatory variables within the newly defined space, often referred to as the "feature space."

b signifies a biased term.

$\in_i$ represent slack variables that indicate the gap or distance between the individual observations (i) and the boundary of the margin associated with their respective classes.

Discovering the ideal hyperplane (see Eq. (8)), which maximizes the margin within the high-dimensional space, is essentially a process of minimizing the norm of the weight vector while also minimizing the count of misclassified instances. Ultimately, the labels or output variables denote the class to which each sample belongs.

$$D(x_i) = W^T \varphi(x_i) + b \tag{8}$$

The scale of the primal model is contingent upon the dimensionality of the problem, whereas the dual form is contingent on the number of samples. Therefore, when the dimensionality is sufficiently high, it becomes more advantageous to address the dual model Eq. (9) to Eq. (11).

$$max_a \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} a_i a_j y_i y_j K(x_i, x_j) \tag{9}$$

$$\sum_{i=1}^{N} a_i y_i = 0 \tag{10}$$

$$0 \leq a_i \leq C_{svc} \qquad i = 1,\ldots,N \tag{11}$$

A Kernel function, denoted as $K(x_i, x_j)$, maps each pair of data points to a corresponding location in the feature space. There are various Kernel functions available, including linear, polynomial, radial basis, sigmoidal, and others. The key requirement for these functions is that they must be symmetric, positive, and semi-definite. Prior research in this field has demonstrated that the radial basis Kernel function, as defined in Eq. (12), is particularly well-suited for classification tasks [29]. Therefore, a radial basis Kernel function is employed with 'γ' serving as a hyperparameter that signifies the inverse of the range of influence of the data points identified as support vectors [30].

$$K(x_i, x_j) = \emptyset(x_i)^R \emptyset(x_j) = exp(-\gamma \|x_j - x_i\|) \tag{12}$$

Once the model has been solved to estimate the weights and the bias term, predictions for new samples can be made using Eq. (13).

$$SVC \quad y_i = \begin{cases} -1 \; if \; w^T \emptyset(x_i) + b \; \leq 0 \\ 1 \; if \; w^T \emptyset(x_i) + b \; > 0 \end{cases} \tag{13}$$

D. Sea Horse Optimization (SHO)

The Sea Horse Optimization (SHO) is a novel metaheuristic inspired by the distinctive behaviors of sea horses [31]. Sea horses display unique mobility patterns, such as periodically wrapping their tails around algal stems in response to oceanic currents and exhibiting Brownian motion-like movements when suspended upside-down. Their specialized head shape enables stealthy predatory approaches with an impressive 90% success rate. Sea horses reproduce through random pairings, allowing their offspring to inherit advantageous traits. These behaviors, encompassing mobility, predatory tactics, and breeding, form the core principles of the SHO algorithm. SHO harnesses the power of swarm intelligence to adapt and optimize solutions, emulating the sea horse's ability to thrive in its environment. This innovative metaheuristic leverages insights from nature to address complex problems efficiently.

The SHO algorithm consists of four key stages, namely, (1) initialization, (2) emulating mobility behavior, (3) simulating predation behavior, and (4) replicating breeding behavior observed in sea horses. Detailed descriptions of each of these stages are provided in the subsequent subsections.

1) Initialization stage: Similar to numerous other metaheuristic algorithms, the SHO commences by initializing the population. In this context, the population of sea horses represents potential problem solutions within the search space, which can be mathematically expressed using Eq. (14):

$$S = \begin{bmatrix} x_1^1 & \dots & x_1^D \\ \dots & \dots & \dots \\ x_P^1 & \dots & x_P^D \end{bmatrix} \tag{14}$$

In Eq. (14), D stands for the variable's dimensionality, P indicates the population's size, and s denotes the sea horses present within the population.

In creating each solution, the problem's upper bound (UB) and lower bound (LB) are employed as initial reference points for random generation. Eq. (15) and Eq. (16) delineate the procedure for generating the $i-th$ individual, denoted as X_i, within the search space [LB, UB].

$$X_i = [x_i^1, \dots, x_i^D] \tag{15}$$

$$x_i^j = rand * (UB^j - LB^j) + LB^j \tag{16}$$

The term $rand$ represents a random number within the range [0, 1]. The variable j is an integer ranging from 1 to D, where D signifies the dimensionality of the problem. The variable i is a positive integer ranging from 1 to P, with P representing the population size. The notation x_i^j refers to the $j-th$ dimension of the i-th individual within the population. The upper and lower bounds for the $j-th$ variable in the optimized problem is denoted as UB^j and LB^j, respectively.

In the context of a minimum optimization problem, the individual with the lowest fitness level is designated as X_{best}, representing the optimal solution. Conversely, in a maximum optimization problem, X_{best} corresponds to the individual with the highest fitness level. The value of X_{best} can be determined using Eq. (17):

$$X_{best} = \arg_{\min\, or\, \max}\left(f(X_i)\right) \tag{17}$$

In the above formula, $f(X_i)$ represents the value of the objective function for a specific task.

2) Movement behavior stage: Sea horses exhibit movement patterns that are akin to a normally distributed random distribution (0,1), resulting in a variety of motion behaviors. To strike a balance between exploiting known information and exploring new possibilities, the algorithm sets a cut-off point at $r_1 = 0$. This means that half of the sea horses are directed towards local search, while the remaining half focus on global exploration. The algorithm's subsequent stages are then employed to manage and further define the motion behavior of these sea horses.

a) First step: The SHO algorithm's exploration strategy is shaped by sea horses' spiral motion, which is affected by oceanic vortexes. When the random value r_1 exceeds the SHO cut-off point, the algorithm prioritizes local exploitation, directing sea horses toward the optimal solution X_{best}. Sea horses move using Lévy flights, promoting exploration in initial iterations and avoiding excessive focus on one area. Their spiral motion includes a continuous adjustment of the rotation angle, widening the search area around local solutions. Eq. (18) is used to generate new positions for sea horses, improving the algorithm's search efficiency.

$$X_{new}^1(t+1) = X_i(t) + Levy\ (\lambda)((X_{best}(t) - X_i(t) * x * y * z + X_{best}(t))$$
$$s.t \begin{Bmatrix} x = p * \cos(\theta) \\ y = p * \sin(\theta) \\ z = p * \theta \\ p = u * e^{\theta v} \end{Bmatrix} \tag{18}$$

u and v are employed to denote the parameters of the logarithmic spiral, which govern the stem length (p). In each case of u and v, a constant of 0.05 is established. The variables x, y, and z represent the three-dimensional coordinates during the spiral motion. θ is chosen randomly from the interval [0, 2π].

Eq. (19) is employed for the computation of the Lévy flight distribution function (Levy(z)):

$$Levy(z) = s * \frac{\omega * \sigma}{|k|^{\frac{1}{\lambda}}} \tag{19}$$

w and k are randomly generated positive values within the range of 0 to 1. The variable s remains constant with a fixed value of 0.01. λ is chosen randomly from the range of 0 to 2, and in this context, it is specifically set to 1.5. The calculation of σ is determined using Eq. (20).

$$\sigma = \left(\frac{\Gamma(1+\lambda) * \sin(\frac{\pi\lambda}{2})}{\Gamma\left(\frac{1+\lambda}{2}\right) * \lambda * 2^{\left(\frac{\lambda-1}{2}\right)}}\right) \tag{20}$$

b) Second step: This algorithm phase portrays sea horses' Brownian motion as a response to oceanic waves. When r_1 falls to the left of the cut-off point, the SHO algorithm transitions into a drifting mode for its search. This shift is vital to avoid trapping the algorithm in local optima.

Brownian motion is employed to mimic sea horses' extended movement, enabling more efficient exploration of the search space. Eq. (21) defines the mathematical representation of this behavior.

β_i denotes the random walk coefficient for the Brownian motion, and l is a constant parameter set at a value of 0.05. The new location of the sea horse at iteration t can be computed by combining the two described situations using Eq. (22).

$$X_{new}^1(t+1) = X_i(t) + rand * l * \beta_t * (X_i(t) - \beta_i * X_{best})\ s.t \left\{\beta_t = \frac{1}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})\right\} \tag{21}$$

$$X_{new}^1(t+1) = \begin{cases} X_i(t) + Levy(\lambda)\left(\left(X_{best}(t) - X_i(t)\right) * x * y * z + X_{best}(t)\right), & r_1 > 0 \\ X_i(t) + rand * l * \beta_t * (X_i(t) - \beta_i * X_{best}), & r_1 \le 0 \end{cases} \tag{22}$$

$$X_{new}^2(t+1) = \begin{cases} \alpha * (X_{best} - rand * X_{new}^1(t) + (1-\alpha) * X_{best}, & if \ \ r_2 > 0.1 \\ (1-\alpha) * (X_{new}^1(t) - rand * X_{best}) + \alpha * X_{new}^1(t), & if \ \ r_2 \le 0.1 \end{cases} \tag{23}$$

Predation Behavior Phase: While sea horses forage for zooplankton and small crustaceans, their predation attempts can result in success or failure. To address this, the SHO algorithm introduces a random variable, r_2, to differentiate these outcomes. With sea horses having a high likelihood of successful hunting (over 90%), the critical threshold for r_2 is set at 0.1. Successful predation in SHO showcases its capability to exploit resources, guided by cues from the best solution's proximity to the prey. A successful predation happens when r_2 exceeds 0.1, leading the sea horse to approach, overtake, and capture the prey (best solution). In the case of unsuccessful predation, both predator and prey reverse their movements, indicating a continuation of exploration. Eq. (23) mathematically depicts this predation behavior.

r_2 denotes a randomly generated integer between 0 and 1, while $X_{new}^1(t)$ shows the novel location of the sea horse after moving at iteration t. The sea horse's movement step size is modified during the pursuit of prey, gradually decreasing with each iteration. This adjustment is computed using Eq. (24).

$$\alpha = \left(1 - \frac{t}{T}\right)^{\frac{2t}{T}} \tag{24}$$

T represents the maximum number of iterations in the algorithm.

3) Breeding behavior phase: In order to accommodate the reproductive patterns of male sea horses, the population is divided into two distinct groups, males and females, categorized according to their fitness levels. Within the framework of the SHO algorithm, the individuals with the most favorable fitness scores constitute the group of chosen fathers, while the rest of the individuals form the group of selected mothers. Eq. (25) illustrates that this partitioning of the population into male and female groups serves the purpose of preventing an over-concentration of novel strategies and encourages the transmission of beneficial traits to both mothers and fathers, ultimately benefiting the subsequent generations.

$$\begin{cases} fathers = X_{sort}^2(1{:}\frac{P}{2}) \\ mothers = X_{sort}^2(\frac{P}{2} + 1{:}p) \end{cases} \tag{25}$$

X_{sort}^2 refers to the collection of all X_{new}^2 solutions organized in ascending order based on their fitness values. In the context of the SHO algorithm, the mothers and fathers correspond to the female and male populations, respectively.

The SHO algorithm operates under the assumption that new offspring are created through the random mating of females and males within the population. In order to maintain the efficiency of the algorithm, it is presumed that each pair of sea horses yields only a single offspring, as evidenced in Eq. (26).

$$X_i^{offspring} = r_3 X_i^{father} + (1 - r_3) X_i^{mother} \tag{26}$$

X_i^{father} and X_i^{mother} denote the male and female members selected at random, respectively. i takes on a positive value within the interval $[1, p/2]$, where p represents another parameter or value within the [0,1], and r_3 is an integer generated at random, which can take values within the range of [0, 1]. Fig. 2 shows the flowchart of SHO.

E. Adaptive Opposition Slime Mould Algorithm (AOSMA)

The Slime Mould Algorithm is based on the oscillatory behavior observed in plasmodial slime mould. This organism utilizes a feedback mechanism that alternates between positive and negative phases along with an oscillatory pattern to determine the best path to obtain nutrients [32]. The Adaptive Opposition Slime Mould Algorithm (AOSMA) is an innovative computational method created to improve the approach behavior of slime mould. It achieves this by incorporating an adaptive decision-making mechanism that is based on opposition-based learning [33].

To construct a mathematical model for the AOSMA, it is assumed that there are a total of "N" individuals belonging to the slime mould species in question residing within the specified search space. This search space is defined by a lower boundary (LB) and an upper boundary (UB).

$X_i = (x_i^1, x_i^2, \cdots, x_i^d), \forall i \in [1, N]$ shows the location of $i-th$ slime mould in d-dimension and $F(X_i), \forall i = [1, N]$ denotes the fitness of the $i-th$ slime.

The following represents the fitness and positions of N slime mould individuals at iteration t:

$$X(x) = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^d \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \tag{27}$$

$$F(X) = [F(X_1), F(X_2), \cdots, F(X_N)\,] \tag{28}$$

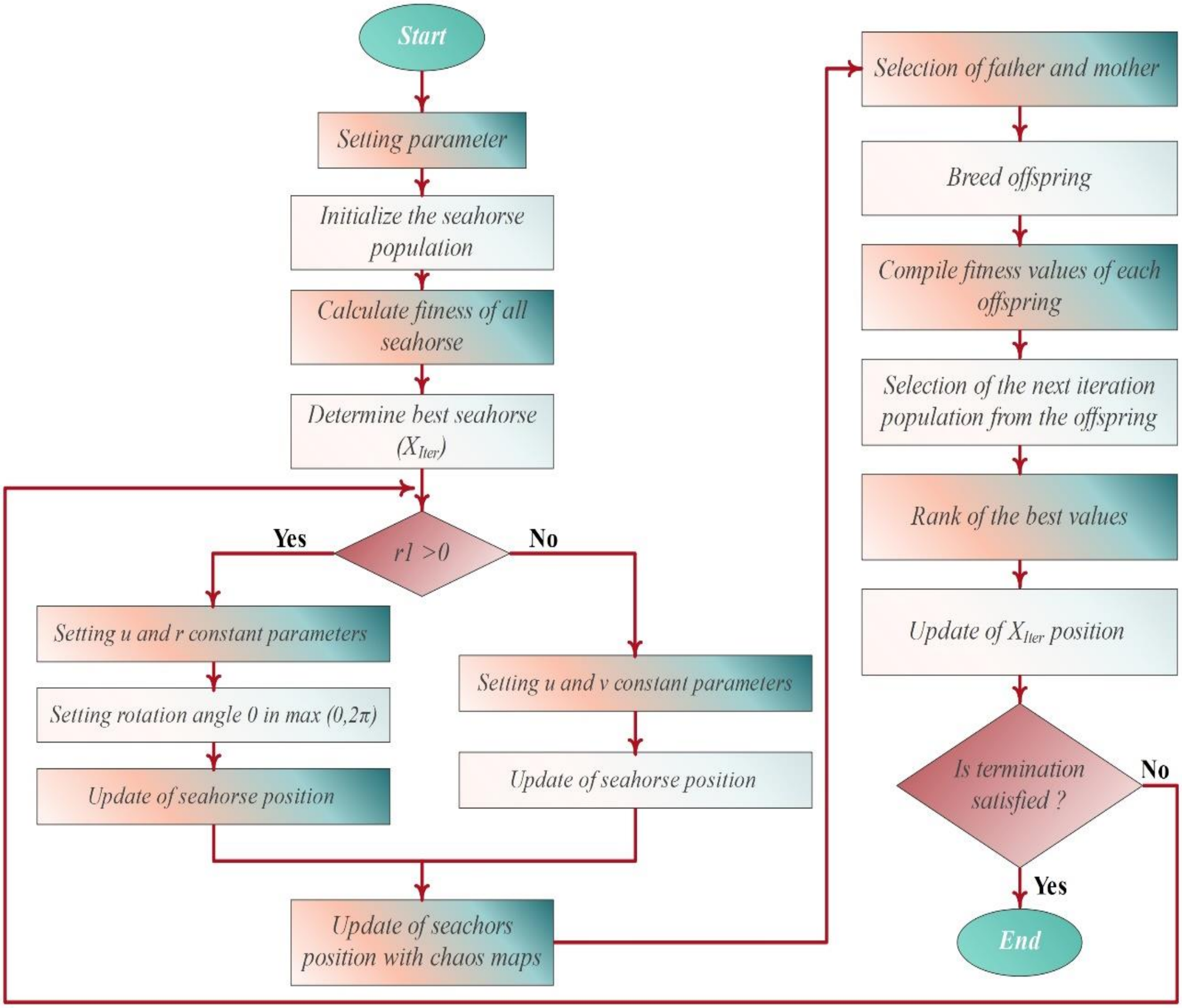


Fig. 2. Flowchart of SHO.

In the iteration at $t+1$, the slime mould's position has progressed, and its spatial arrangement has been enhanced as determined by Eq. (29):

$$X_i(t+1) = \begin{cases} X_{LB}(t) + V_d\big(W.X_A(t) - X_B(t)\big) & p_1 \geq \delta \text{ and } p_2 < m_i \\ V_e.X_i(t) & p_1 \geq \delta \text{ and } p_2 \geq m_i, \forall i \in [1,N] \\ rand.(UB - LB) + LB & p_1 < z \end{cases} \quad (29)$$

X_{LB} represents the top-performing local slime mould while X_A and X_B denote individuals selected at random. The equation incorporates a weight factor (W), along with random velocities (V_d and V_e), as well as two randomly chosen values, p_1 and p_2, within the range [0, 1]. The fixed value of $\delta = 0.03$ signifies the slime mould's initial chance to explore a random search location. Additionally, m_i represents the threshold value for the i-th member in the population, which aids in determining the position of the slime mould, and this is computed according to Eq. (30) to Eq. (32)

$$m_i = tanh|F(X_i) - F_G|, \forall i \in [1,N] \quad (30)$$

$$F_G = F(X_G) \quad (31)$$

$$W\big(SortInd_F(i)\big) = \begin{cases} 1 + rand.\log\left(\frac{F_{LB} - F(X_i)}{F_{LB} - F_{Lw}} + 1\right) & 1 \leq i \leq \frac{N}{2} \\ 1 - rand.\log\left(\frac{F_{LB} - F(X_i)}{F_{LB} - F_{Lw}} + 1\right) & \frac{N}{2} < i \leq N \end{cases} \quad (32)$$

$rand$ signifies a randomly generated number within the range of 0 to 1. F_{LB} and F_{Lw} represent the fitness values corresponding to the local best and worst outcomes, while F_G and X_G stand for the global best fitness value and the associated global best position, respectively.

Sorting fitness values in ascending order can be used when dealing with a minimization problem.

$$[Sort_F, SortInd_F\ ] = sort(F) \quad (33)$$

The local best fitness values, as well as the local best slime mould X_{LB}, are calculated using Eq. (34-36).

$$F_{LB} = F\big(Sort_F(1)\big) \quad (34)$$

$$F_{LW} = F\big(Sort_F(N)\big) \quad (35)$$

$$X_{LB} = X\big(SortInd_F(1)\big) \quad (36)$$

The random velocities are denoted as V_d and V_e, are defined as follows:

$$V_d \in [-d, d] \quad (37)$$

$$V_e \in [-e, e] \quad (38)$$

$$d = \text{arctanh}\left(-\left(\frac{t}{T}\right) + 1\right) \quad (39)$$

$$e = 1 - \frac{t}{T} \quad (40)$$

In the context of engineering design problem-solving and optimization, the Slime Mould Algorithm (SMA) demonstrates significant potential for both exploration and exploitation. The enhancement of slime mould rules within the SMA framework is contingent on several key scenarios.

Case 1: When $p_1 \geq z$ and $p_2 < m_i$, the search guided by the local best slime mould X_{LB} and two random individuals X_A and X_B with velocity V_d. This step facilitates the achievement of a balance between the activities of exploitation and exploration.

Case 2: When $p_1 \geq z$ and $p_2 \geq m_i$, the search is guided by the position of slime mould with a velocity V_e. This case assists in exploitation.

Case 3: When $p_1 < z$, the individual reinitializes in a defined search space. This step helps in exploration.

Case 1 illustrates that as X_A and X_B are two random slime moulds, the chances of obtained solutions are not managed properly in exploration and exploitation. To overcome this shortcoming, local best individual X_{LB} can be replaced by X_A. Therefore, the $i-th$ member's position is remodeled as Eq. (41):

$$Xn_i(t) = \begin{cases} X_{LB}(t) + V_d\big(W.X_{LB}(t) - X_B(t)\big) & p_1 \geq \delta \ and\ p_2 < m_i \\ V_e.X_i(t) & p_1 \geq \delta \ and\ p_2 \geq m_i \\ rand.(UB - LB) + LB & p_1 < \delta \end{cases} \quad (41)$$

Case 2 elucidates that slime mould strategically capitalizes on a locale in its vicinity, thereby resorting to a trajectory characterized by a diminished level of fitness. To address this issue, implementing an adaptive decision mechanism presents a superior solution.

Case 3 highlights that while the Slime Mould Algorithm (SMA) supports exploration, a low δ value of 0.03 limits this aspect. To address this, introducing an auxiliary exploration component is crucial. An effective strategy involves using opposition-based learning (OBL) to determine when additional exploration is needed [34]. OBL utilizes a specific Xop_i in the search space opposite to Xni for each member, improving convergence and preventing local minima traps. Xop_i for the $i-th$ individual in the $j-th$ dimension ($j = 1,2,\cdots,s$) is defined accordingly. This adaptation resolves the limitations of Cases 2 and 3 in SMA.

$$Xop_i^j = \min\big(Xn_i(t)\big) + max\big(Xn_i(t)\big) - Xn_i^j(t) \quad (42)$$

The position of the $i-th$ member in the minimization problem is denoted as Xr_i, is defined as follows:

$$Xr_i = \begin{cases} Xop_i(t) & F(Xop_i(t)) < F\big(Xn_i(t)\big) \\ Xn_i(t) & F(Xop_i(t)) \geq F\big(Xn_i(t)\big) \end{cases} \quad (43)$$

An adaptive decision hinges on both the previous fitness value, $f(Xi(t))$, and the current fitness value, $f(Xni(t))$, when a nutrient pathway is exhausted. This scholarly writing style supports the need for further research and, as a result, improves the position for the next iteration in the following manner:

$$X_i(t+1) = \begin{cases} Xn_i(t) & F(Xn_i(t)) \leq F\big(X_i(t)\big) \\ Xr_i(t) & F(Xn_i(t)) > F\big(X_i(t)\big) \end{cases}, \quad \forall i \in [1, N] \quad (44)$$

The AOSMA algorithm described above is presented in pseudocode, as depicted in Algorithm 1.

Algorithm 1: AOSMA Algorithm

Begin
Inputs: N, s, T, δ and select an objective function f with search boundary range $[LB, UB]$.
Outputs: X_G and F_G
Initialization: Randomly initialize the slime mould $X_i = (x_i^1, x_i^2, \cdots, x_i^d)$, $\forall i \in [1, N]$ within the search boundary UB and LB for initial iteration $t = 1$.
while ($t \leq T$)
Calculate the fitness values $F(X)$ of N slime mould.
Sort the fitness value.
Update the local best fitness F_{LB} corresponding local best individual X_{LB}.
Update the local worst fitness F_{LW}.
Update the global best fitness F_G and corresponding global best individual X_G.
Update the weight W.
Update the d
for (each slime mould $i = 1{:}N$)
Generate random numbers p_1 and p_2.
Generate the threshold value m_i.
Evaluate new slime mould position Xn_i
Evaluate the fitness value of the new slime mould F(Xn_i).
if ($F(Xn_i) > F(X_i)$ // Adaptive decision strategy
Estimate Xop_i. //Opposition-based learning
Select Xr_i
End
Update the next iteration slime mould X_i
end
Next iteration $t = t + 1$
end
Return: Global best solution space X_G.

IV. Results and Discussion

A. Convergence Results

The SHO and AOSMA, two potent metaheuristic optimization algorithms, were used in this work to optimize and fine-tune the SVC model's hyperparameters, especially the hybrid models SVSH and SVAO. Improving these algorithms' prediction accuracy was the main goal. A convergence curve, measuring accuracy over 200 iterations, was used to assess the convergence of different optimization techniques, as shown in Fig. 3. This curve allowed for the evaluation of convergence progress and rate by providing a visual representation of the accuracy progression with each repetition. Although the convergence rates of the SVSH and SVAO models were originally comparable, the SVSH model eventually attained a better degree of accuracy. Interestingly, the trend line's linear shape at the 150-iteration mark revealed the ideal computing efficiency threshold for both models. SVSH showed better prediction accuracy throughout the optimization phase. This study used SHO and AOSMA to improve SVC models.

B. Comparing Results of Predictive Models

Three prediction models were created in this study using a categorization technique to forecast students' test performance and gradually improve their future grades. The models included two others and a single Support Vector Classification (SVC) that were improved with the help of the Adaptive Opposition Slime Mould Algorithm (AOSMA) and the Sea Horse Optimization (SHO). Thirty percent of the dataset was used for test, while the remain seventy percent was used for train. For every model, Accuracy, Recall, Precision, and F1-score are shown in Table I for the training and testing stages, and Fig. 4 illustrates these values. Higher metric values during train than during test indicated that SVSH outperformed the other models in terms of train performance. The $maximum$ metric values achieved by SVSH were $Accuracy = 0.924, Precision = 0.930, Recall = 0.920, and\ F1 - score = 0.920$. In contrast, the SVC model obtained the lowest values, with $Accuracy = 0.887, Precision = 0.89, Recall = 0.89, and\ F1 - Score = 0.89$.

Based on test results (G3 values), an in-depth analysis of 649 students was carried out after data processing and a thorough assessment of the models' categorization skills throughout both the training and testing stages. These pupils were divided into four groups: Poor (which included pupils with G3 scores between 0 and 12), Acceptable (which included pupils with G3 scores between 12 and 14), Good (which included pupils with G3 scores between 14 and 20), and Excellent (which included pupils with G3 scores between 16 and 20). 82 pupils were placed in the Excellent category, 112 in the Good category, 154 in the Acceptable category, and 301 in the Poor category as a consequence of this classification. The findings of this research indicate that 46.38% of students had low academic achievement, with the remaining pupils displaying acceptable, good, and exceptional educational performance, at 23.73%, 17.26%, and 12.63%, respectively. The recall, precision, and F1-score Index values are shown in Table II and are used as assessment metrics to gauge how well the constructed models perform in terms of categorization across different student groups. A comparison study that considers each of these three Index values is presented in the next section.

1) Precision: A thorough evaluation of two refined models showed that the SVSH model had the greatest values in the Good and Poor groups when it came to student classification, with accuracy scores of 0.88 and 0.97, respectively. On the other hand, for the Acceptable group, the SVAO model produced a maximum precision value of 0.88. With an accuracy score of 0.97, the SVC model fared better than the others for the Excellent category.

2) Recall: The SVSH model showed the greatest recall values in the Acceptable and Excellent categories, with scores of 0.9 and 0.88, respectively. In contrast, the Good group's SVAO model achieved a maximum accuracy value of 0.89. The SVC model performed the best for the Poor group, with a recall score of 0.98.

3) F1-score: An improved F1-score indicates that the model can balance accurately detecting positive instances (precision) with including all true positive cases (recall). When all student categories are taken into account, the SVSH model performs very well, as seen by F1-scores of 0.92, 0.88, 0.88, and 0.97 for students who are categorized as Excellent, Good, Acceptable, and Poor.

In summary, when analyzing the complete dataset, the SVSH model unequivocally emerges as the top-performing predictor among all the models.

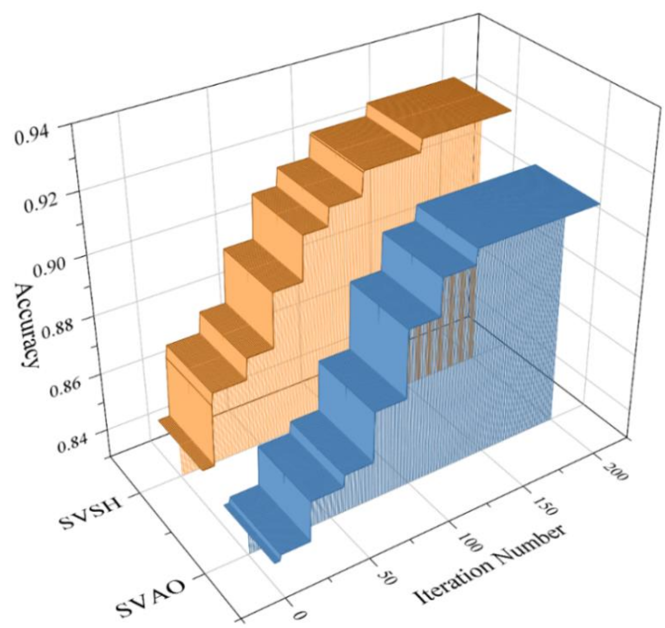


Fig. 3. Convergence curve of hybrid models.

TABLE I. RESULT OF PRESENTED MODELS

Model	Phase	Index values			
		Accuracy	*Precision*	*Recall*	*F1 − Score*
SVC	*Train*	0.904	0.900	0.900	0.900
	Test	0.887	0.890	0.890	0.880
	All	0.904	0.900	0.900	0.900
SVSH	*Train*	0.924	0.930	0.920	0.920
	Test	0.877	0.880	0.880	0.880
	All	0.924	0.930	0.920	0.920
SVAO	*Train*	0.912	0.910	0.910	0.910
	Test	0.892	0.890	0.890	0.890
	All	0.912	0.910	0.910	0.910

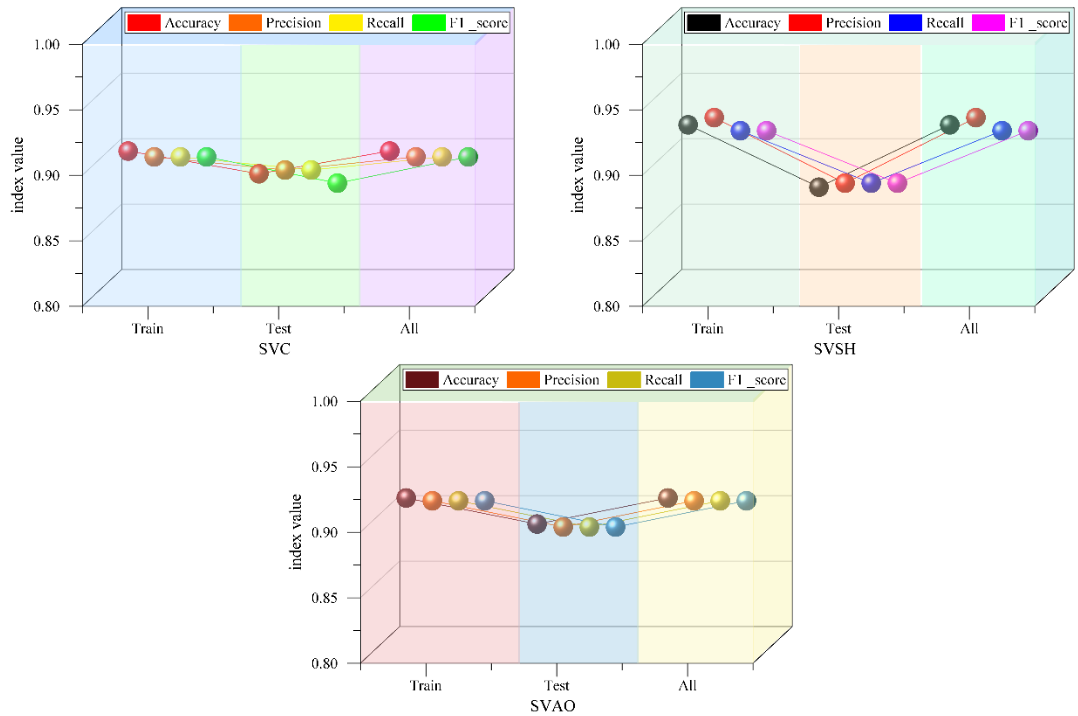
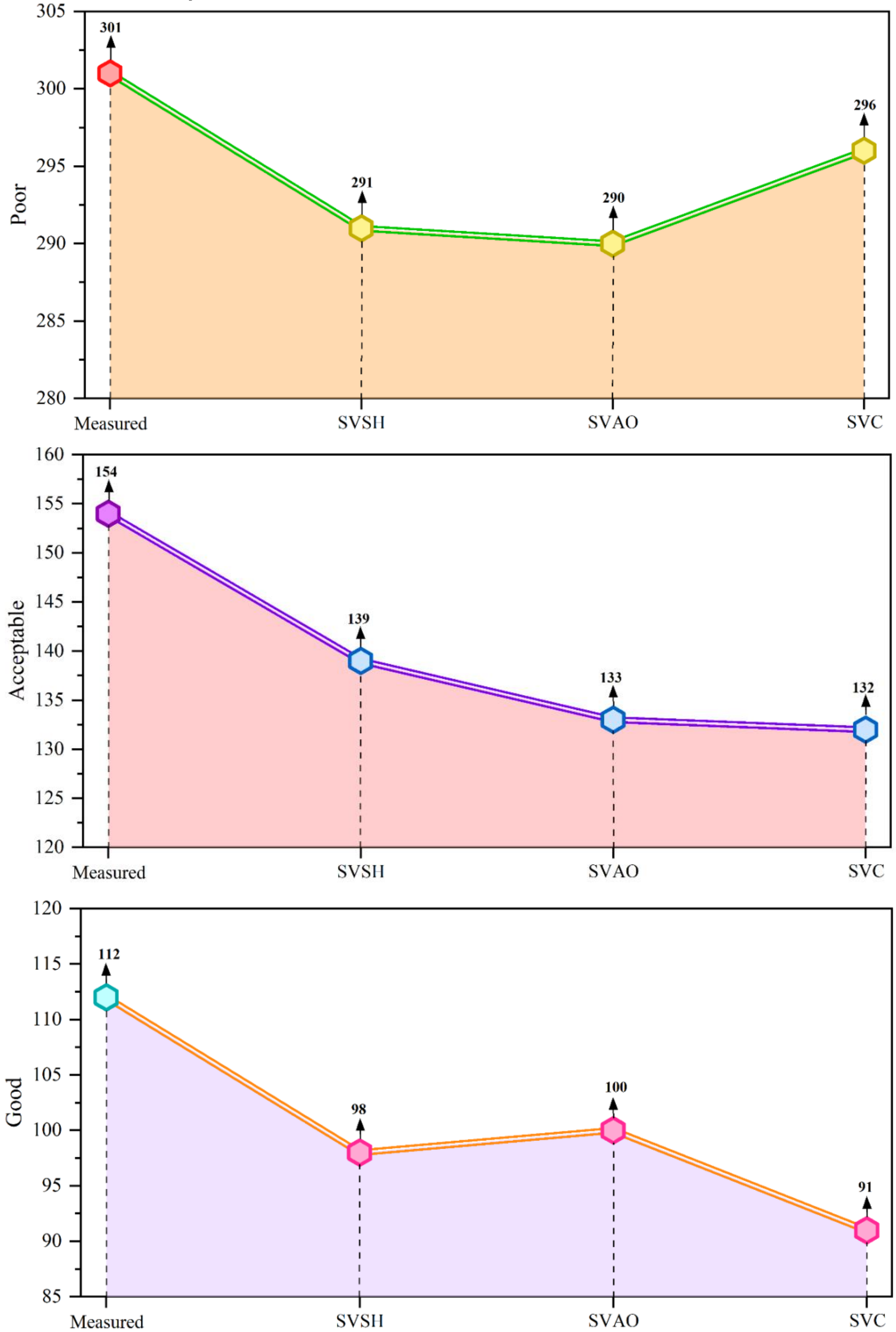


Fig. 4. Models' prediction performance.

TABLE II. EVALUATION INDEXES OF THE DEVELOPED MODELS' PERFORMANCE BASED ON GRADES

Model	Grade	Index values		
		Precision	*Recall*	*F1 − score*
SVC	*Excellent*	0.970	0.830	0.890
	Good	0.830	0.810	0.820
	Acceptable	0.860	0.860	0.860
	Poor	0.930	0.980	0.960
SVSH	*Excellent*	0.960	0.880	0.920
	Good	0.880	0.880	0.880
	Acceptable	0.860	0.900	0.880
	Poor	0.970	0.970	0.970
SVAO	*Excellent*	0.960	0.840	0.900
	Good	0.810	0.890	0.850
	Acceptable	0.880	0.860	0.870
	Poor	0.960	0.960	0.960

There were, in fact, 301, 154, 112, and 82 kids in the Poor, Acceptable, Good, and Excellent categories. To facilitate a visual comparison, Fig. 5 presents the student distribution across these categories in a visual manner based on the results of the measurement and classification models. It is noteworthy that the SVSH model successfully classified 139 and 72 students into the Acceptable and Excellent categories, respectively, with the maximum accuracy. Classifying 296 students properly, the SVC model outperformed the other models in the Poor group. Finally, the SVAO model worked best for the Good group, correctly classifying 100 pupils.



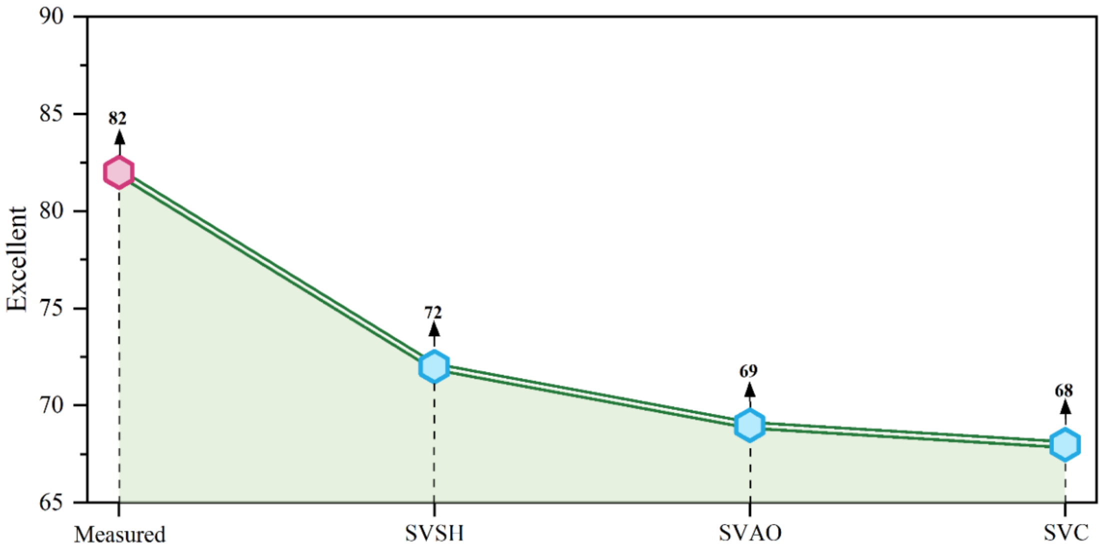


Fig. 5. Symbol-line drop plot based on measured and classification models' outcomes.

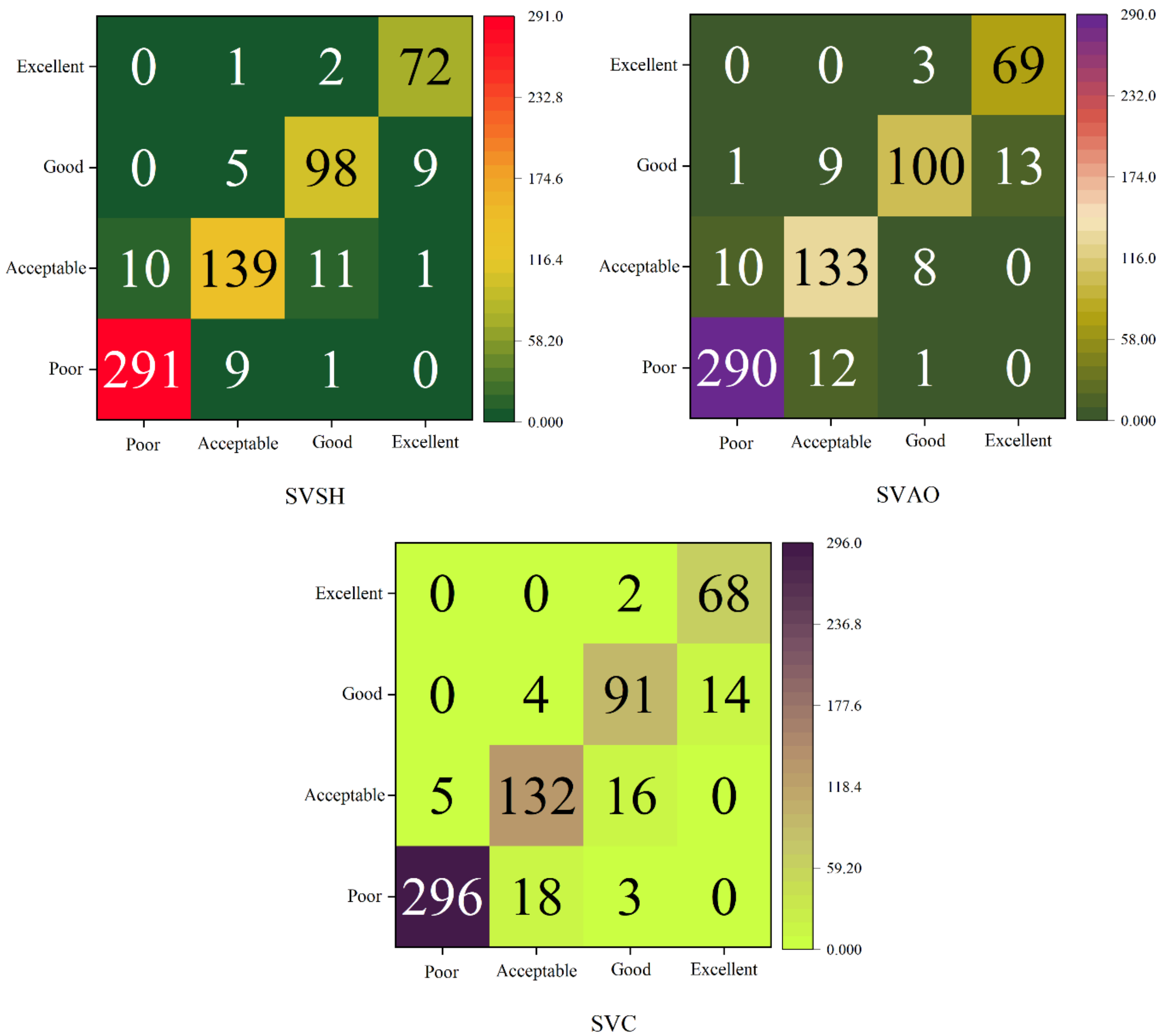


Fig. 6. Confusion matrix for each model's accuracy.

The Fig. 6 confusion matrix offers valuable information on both the proper placement of pupils in their corresponding grades and their incorrect classification into unrelated groups. In particular, just 49 students were misclassified when using the SVSH model, which properly placed 72, 98, 139, and 291 students in the Excellent, Good, Acceptable, and Poor courses, respectively. However, 57 and 62 pupils, respectively, were incorrectly categorized by the SVAO and SVC models. Notably, the two optimized models mostly misclassified students between surrounding categories. For example, students 9 and 13 for SVSH and SVAO were incorrectly put in the good group rather than the Excellent category. Three pupils were mistakenly assigned to the good category in the single SVC model, rather than the poor group. In summary, SVSH demonstrated higher predictive accuracy than the other two models in predicting students' future academic achievement.

V. Discussion

A. Comparison

Table III compares the accuracy of different models across studies. Pallathadka et al. [35] achieved 89% accuracy with SVM and 78% with NB. Shreem et al. [36] obtained 87% accuracy with NB. In the present study, the SVSH model achieved the highest accuracy at 92.4%, surpassing the results from previous studies. This highlights the effectiveness of the SVSH model in student performance prediction, showcasing its potential for improved accuracy compared to SVM and NB models in the referenced research.

TABLE III. COMPARISON WITH PUBLISHED PAPERS

Article	Model	Accuracy
Pallathadka et al. [35]	SVM	89%
	NB	78%
Shreem et al. [36]	NB	87%
Present study	SVSH	92.4%

VI. Conclusion

This research emphasizes the critical role that data-driven prediction models play in the field of education, stressing the value of combining quantitative and qualitative elements in the process of predicting and evaluating the academic achievement of students. It offers insightful information that will help students, academic institutions, and legislators drive future advancements in education. The study demonstrates how well data mining methods like regression, clustering, and classification work to understand and proactively handle the variety of problems college students experience. Furthermore, by combining the Support Vector Classification (SVC) model with optimization methods like Sea Horse Optimization (SHO) and the Adaptive Opposition Slime Mold Algorithm (AOSMA), the work presents a novel methodology. This innovative approach shows how optimization algorithms and sophisticated machine learning methods may improve the accuracy and efficacy of prediction models, providing a potent toolset for tackling the changing obstacles faced by students across their academic careers. It is clear from a thorough assessment approach that involves splitting the models into train and test sets that these hybrid models have the ability to greatly improve the SVC model's classification skills. The accuracy and precision are significantly improved by this addition.

Notably, the SVSH fared better than the SVAH, scoring around 2% higher in accuracy and precision.Furthermore, the SHO's success in improving classification accuracy was notable when 649 students were classified based on their final grades. With an astounding accuracy rate of 92.45%, the SVSH model in particular showed remarkable capacity to correctly categorize the majority of pupils. By comparison, 8.78% and 9.55% of all pupils were incorrectly categorized by SVAO and SVC, respectively. This demonstrates the SVSH model's higher prediction ability in correctly classifying pupils according to their final grades.

Funding

This work was supported by Guangdong Provincial Philosophy and Social Sciences Planning 2023 Discipline Co-construction Project "Research on the Cultivation Model of New Professional Farmers in East Guangdong in the New Era"(GD23XJY82); 2023 Guangdong Province Education Science Planning Project (Higher Education Special) "Application Limits of Artificial Intelligence in Higher Education and Ethical Risk Assessment Research" (2023GXJK918).

References

[1] C. Romero and S. Ventura, “Educational data mining: A survey from 1995 to 2005,” Expert Syst Appl, vol. 33, no. 1, pp. 135–146, 2007.

[2] Y. Ma, B. Liu, C. K. Wong, P. S. Yu, and S. M. Lee, “Targeting the right students using data mining,” in Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000, pp. 457–464.

[3] D. Kabakchieva, K. Stefanova, and V. Kisimov, “Analyzing university data for determining student profiles and predicting performance,” in Educational Data Mining 2011, 2010.

[4] R. S. J. D. Baker and K. Yacef, “The state of educational data mining in 2009: A review and future visions,” Journal of educational data mining, vol. 1, no. 1, pp. 3–17, 2009.

[5] R. S. J. D. Baker and K. Yacef, “The state of educational data mining in 2009: A review and future visions,” Journal of educational data mining, vol. 1, no. 1, pp. 3–17, 2009.

[6] E. Chandra and K. Nandhini, “Knowledge mining from student data,” European journal of scientific research, vol. 47, no. 1, pp. 156–163, 2010.

[7] A. Ahmed and I. S. Elaraby, “Data mining: A prediction for student’s performance using classification method,” World Journal of Computer Application and Technology, vol. 2, no. 2, pp. 43–47, 2014.

[8] M. M. A. Tair and A. M. El-Halees, “Mining educational data to improve students’ performance: a case study,” International Journal of Information, vol. 2, no. 2, 2012.

[9] behnam Sedaghat, G. G. Tejani, and S. Kumar, “Predict the Maximum Dry Density of soil based on Individual and Hybrid Methods of Machine Learning,” Advances in Engineering and Intelligence Systems, vol. 002, no. 03, 2023, doi: 10.22034/aeis.2023.414188.1129.

[10] H. A. A. Hamza and P. Kommers, “A review of educational data mining tools & techniques,” International Journal of Educational Technology and Learning, vol. 3, no. 1, pp. 17–23, 2018.

[11] C. Romero and S. Ventura, “Educational data mining: a review of the state of the art,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (applications and reviews), vol. 40, no. 6, pp. 601–618, 2010.

[12] C. Márquez-Vera, A. Cano, C. Romero, and S. Ventura, “Predicting student failure at school using genetic programming and different data

mining approaches with high dimensional and imbalanced data," Applied intelligence, vol. 38, pp. 315–330, 2013.

[13] C. Márquez-Vera, A. Cano, C. Romero, and S. Ventura, "Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data," Applied intelligence, vol. 38, pp. 315–330, 2013.

[14] D. Kabakchieva, "Student performance prediction by using data mining classification algorithms," International journal of computer science and management research, vol. 1, no. 4, pp. 686–690, 2012.

[15] E. Osmanbegovic and M. Suljic, "Data mining approach for predicting student performance," Economic Review: Journal of Economics and Business, vol. 10, no. 1, pp. 3–12, 2012.

[16] S. Huang and N. Fang, "Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models," Comput Educ, vol. 61, pp. 133–145, 2013.

[17] A. K. Pal and S. Pal, "Data mining techniques in EDM for predicting the performance of students," International Journal of Computer and Information Technology, vol. 2, no. 06, pp. 764–2279, 2013.

[18] L. Ramanathan, S. Dhanda, and D. S. Kumar, "Predicting students' performance using modified ID3 algorithm," International Journal of Engineering and Technology, vol. 5, no. 3, pp. 2491–2497, 2013.

[19] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition," Expert Syst Appl, vol. 41, no. 2, pp. 321–330, 2014.

[20] Y.-H. Hu, C.-L. Lo, and S.-P. Shih, "Developing early warning systems to predict students' online learning performance," Comput Human Behav, vol. 36, pp. 469–478, 2014.

[21] S. Natek and M. Zwilling, "Student data mining solution–knowledge management system related to higher education institutions," Expert Syst Appl, vol. 41, no. 14, pp. 6400–6407, 2014.

[22] F. Marbouti, H. A. Diefes-Dux, and K. Madhavan, "Models for early prediction of at-risk students in a course using standards-based grading," Comput Educ, vol. 103, pp. 1–15, 2016.

[23] M. Pandey and S. Taruna, "A multi-level classification model pertaining to the student's academic performance prediction," Int J Adv Eng Technol, vol. 7, no. 4, p. 1329, 2014.

[24] E. B. Costa, B. Fonseca, M. A. Santana, F. F. de Araújo, and J. Rego, "Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses," Comput Human Behav, vol. 73, pp. 247–256, 2017.

[25] P. Cortez and A. M. G. Silva, "Using data mining to predict secondary school student performance," 2008.

[26] V. Vapnik, "Statistical Learning Theory. New York: John Willey & Sons," Inc, 1998.

[27] S. Maldonado, J. Pérez, R. Weber, and M. Labbé, "Feature selection for support vector machines via mixed integer linear programming," Inf Sci (N Y), vol. 279, pp. 163–175, 2014.

[28] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM transactions on intelligent systems and technology (TIST), vol. 2, no. 3, pp. 1–27, 2011.

[29] M. Aydogdu and M. Firat, "Estimation of failure rate in water distribution network using fuzzy clustering and LS-SVM methods," Water resources management, vol. 29, pp. 1575–1590, 2015.

[30] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.," 2022.

[31] S. Zhao, T. Zhang, S. Ma, and M. Wang, "Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems," Applied Intelligence, vol. 53, no. 10, pp. 11833–11860, 2023, doi: 10.1007/s10489-022-03994-3.

[32] M. K. Naik, R. Panda, and A. Abraham, "Adaptive opposition slime mould algorithm," Soft comput, vol. 25, no. 22, pp. 14297–14313, 2021.

[33] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," Future Generation Computer Systems, vol. 111, pp. 300–323, 2020.

[34] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), IEEE, 2005, pp. 695–701.

[35] 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," Mater Today Proc, vol. 80, pp. 3782–3785, 2023.

[36] S. S. Shreem, H. Turabieh, S. Al Azwari, and F. Baothman, "Enhanced binary genetic algorithm as a feature selection to predict student performance," Soft comput, vol. 26, no. 4, pp. 1811–1823, 2022.