The Scientific Temper (2024) Vol. 15 (spl): 121-131
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.15
https://scientifictemper.com/
RESEARCH ARTICLE

Feature selection in HR analytics: A hybrid optimization approach with PSO and GSO

Jayalakshmi K.*, M. Prabakaran

Abstract

In the field of Human Resources (HR) analytics, effective feature selection is critical for improving the accuracy and efficiency of predictive models used for workforce management, talent retention, and performance evaluation. This paper proposes an improved feature selection approach that integrates optimization techniques such as particle swarm optimization (PSO) and gravitational search optimization (GSO) to enhance the performance of HR analytics. By leveraging the exploration-exploitation balance of PSO and the mass-based search capability of GSO, the proposed method efficiently identifies the most relevant features from large and complex HR datasets. The hybrid approach reduces dimensionality, minimizes computational costs, and boosts the accuracy of machine learning models used in HR analytics. Comparative analysis with traditional feature selection methods demonstrates that the proposed technique achieves superior results in terms of prediction accuracy, computational efficiency, and overall model performance. This study highlights the potential of advanced optimization techniques in driving data-driven decision-making processes in HR, offering a robust and scalable solution for managing and analyzing HR data more effectively.

Keywords: HR analytics, Big data, Feature selection, Classification, Particle swarm optimization, Gravitational search optimization.

Introduction

Human resource (HR) analytics, also known as people analytics or talent analytics, is transforming how organizations manage and optimize their workforce. Traditionally, HR departments have focused on tasks like recruitment, payroll, and compliance, operating with limited involvement in strategic decision-making. However, the rapid digitalization and availability of workforce data have shifted HR's role, enabling it to become a key driver of organizational performance. By systematically collecting, analyzing, and interpreting employee data, HR analytics provides insights that help organizations make data-driven decisions to enhance recruitment, improve employee retention, foster engagement, and drive overall productivity, Navot, A. (2006).

HR analytics leverages a wide range of data sources, from internal records—such as employee performance, turnover rates, and training history—to external data like labor market trends and industry benchmarks. Using data mining, statistical analysis, and machine learning techniques, HR analytics reveals patterns, relationships, and trends that may otherwise go unnoticed. For example, predictive analytics can help organizations identify employees who might be at risk of leaving, while prescriptive analytics can provide actionable strategies to mitigate these risks. These insights empower HR departments to make proactive, rather than reactive, decisions, thus aligning HR strategy with broader business goals, Durairaj, M., & Poornappriya, T. S. (2020).

Importance of Feature Selection Techniques

Feature selection techniques play a crucial role in improving the classification accuracy of healthcare decision support systems by enhancing the quality of input data and reducing the dimensionality of feature space, Dash, M., & Liu, H. (1997), Nourmohammadi-Khiarak, J., Feizi-Derakhshi, M. R., Behrouzi, K., Mazaheri, S., Zamani-Harghalani, Y., & Tayebi, R. M. (2020).

Improved Model Performance

Healthcare decision support systems often deal with high-dimensional data containing numerous features, some

PG & Research Department of Computer Science, Government Arts College (Autonomous) (Affiliated to Bharathidasan University, Tiruchirappalli), Karur, Tamilnadu, India.

***Corresponding Author:** Jayalakshmi K., PG & Research Department of Computer Science, Government Arts College (Autonomous) (Affiliated to Bharathidasan University, Tiruchirappalli), Karur, Tamilnadu, India., E-Mail: jaya.senthamarai@gmail.com

How to cite this article: Jayalakshmi, K., Prabakaran, M. (2024). Feature selection in HR analytics: A hybrid optimization approach with PSO and GSO. The Scientific Temper, **15**(spl):121-131.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.spl.15

Source of support: Nil

Conflict of interest: None.

Received: 06/09/2024 **Accepted:** 19/09/2024 **Published**: 16/10/2024

of which may be irrelevant, redundant, or noisy. Feature selection helps in identifying the most informative and relevant subset of features, allowing machine learning models to focus on the most discriminative information. This results in improved model performance and better classification accuracy.

Reduced Overfitting

Including irrelevant or redundant features in the model can lead to overfitting, where the model captures noise or patterns specific to the training data but fails to generalize well to unseen data. Feature selection mitigates overfitting by excluding irrelevant or redundant features, thereby improving the model's ability to generalize to new data instances and reducing the risk of erroneous predictions.

Enhanced Interpretability

In healthcare decision support systems, interpretability is often as important as accuracy, especially when providing explanations for clinical decisions. Feature selection helps in simplifying the model by selecting a smaller subset of features, which enhances the interpretability of the model and facilitates understanding by healthcare professionals and end-users.

Faster Model Training and Inference

Reducing the dimensionality of the feature space through feature selection leads to faster model training and inference times. With a smaller subset of features, computational resources are utilized more efficiently, resulting in quicker decision-making processes in real-time healthcare scenarios, where timely responses are crucial.

Mitigation of Data Imbalance

Healthcare datasets often suffer from class imbalance, where one class (e.g., rare diseases) is significantly underrepresented compared to others. Feature selection can help alleviate data imbalance by selecting features that are more informative for distinguishing between different classes, thereby improving the classification performance for minority classes.

Identification of Biomarkers and Risk Factors

Feature selection techniques enable the identification of biomarkers and risk factors associated with specific diseases or medical conditions. By selecting relevant features from heterogeneous healthcare data sources (e.g., electronic health records, medical imaging), feature selection aids in uncovering meaningful patterns and associations that contribute to better diagnosis, prognosis, and treatment planning.

Feature Selection Techniques

Feature selection techniques are methods used to identify the most relevant subset of features from a larger set of features. These techniques are widely employed in various fields, including machine learning, data mining, and bioinformatics, to improve model performance, reduce dimensionality, and enhance interpretability.

Filter Methods

Filter-based feature selection methods operate independently of any specific machine learning algorithm and are typically based on statistical measures or scoring functions to rank or evaluate the relevance of features. These methods are computationally efficient and can be applied as a preprocessing step before model training, Siddiqi, M. A., & Pak, W. (2020), Jha, K., & Saha, S. (2021).

Correlation-based Feature Selection

Correlation-based feature selection assesses the relationship between each feature and the target variable or between pairs of features. For classification tasks, features with high correlation to the target variable are deemed informative and are retained, while irrelevant features are discarded. In cases of multicollinearity (high correlation between features), only one of the correlated features may be retained to avoid redundancy. Pearson correlation coefficient, Spearman rank correlation coefficient, or other similar metrics are commonly used to quantify the degree of correlation.

Information Gain

Information gain measures the reduction in entropy (uncertainty) of the target variable given the presence of a particular feature. Features that result in a significant reduction in entropy when included in the model are considered more informative and are selected. This method is particularly popular in decision tree-based algorithms, where features are selected based on their ability to split the dataset into pure subsets (homogeneous with respect to the target variable).

Chi-square Test

The chi-square test assesses the independence between each feature and the target variable in classification tasks.

It calculates the chi-square statistic for each feature, which measures the difference between the observed and expected frequencies of each class label given the presence or absence of the feature. Features with high chi-square statistics (indicating significant association with the target variable) are retained, while features with low statistics are discarded.

Variance Thresholding

Variance thresholding removes features with low variance, assuming that features with little variation across instances contribute little information to the model. This method is particularly useful for datasets where most instances have the same value for a feature, as these features do not discriminate between instances and can be safely removed. Users can specify a threshold value below which features are considered low-variance and are subsequently discarded.

Mutual Information

Mutual information measures the mutual dependence between two random variables (e.g., feature and target variable) and quantifies the amount of information obtained about one variable through the other. In feature selection, mutual information assesses the amount of information shared between each feature and the target variable. Features with high mutual information are considered more informative and are retained, while features with low mutual information are discarded.

Wrapper Methods

Wrapper-based feature selection methods assess the quality of feature subsets by directly evaluating their performance using a specific machine learning algorithm. These methods involve iterative search strategies to explore the space of possible feature subsets and an evaluation metric to assess the performance of each subset, Le, T. M., Vo, T. M., Pham, T. N., & Dao, S. V. T. (2020), Ghosh, M., Guha, R., Sarkar, R., & Abraham, A. (2020).

Forward Selection

Forward selection starts with an empty set of features and iteratively adds one feature at a time based on their individual performance. At each iteration, the algorithm evaluates the performance of all possible feature combinations by training a model with each combination and selecting the one that yields the best performance according to a predefined evaluation metric. The process continues until a stopping criterion is met, such as reaching a specified number of features or no improvement in performance. Forward selection tends to be computationally expensive, especially for datasets with a large number of features, as it requires training multiple models for each iteration.

Backward Elimination

Backward elimination starts with the full set of features and iteratively removes the least important feature based on their performance. At each iteration, the algorithm evaluates the performance of the current feature subset by training a model and then removes the feature that contributes the least to the model's performance according to a predefined evaluation metric. The process continues until a stopping criterion is met, such as reaching a specified number of features or no improvement in performance. Backward elimination tends to be more computationally efficient than forward selection, especially for datasets with a large number of features, as it involves training fewer models.

Recursive Feature Elimination (RFE)

Recursive feature elimination (RFE) is a wrapper-based feature selection method that selects features by recursively considering smaller and smaller feature subsets until the desired number of features is reached. RFE starts with the full set of features and trains a model to rank the importance of each feature. At each iteration, the least important feature(s) are removed from the current feature subset, and the process repeats until the desired number of features is reached. The performance of the feature subset is evaluated using a predefined evaluation metric at each iteration, and the subset with the best performance is selected. RFE is particularly useful when the number of features is much larger than the number of instances, as it helps prevent overfitting by iteratively eliminating irrelevant features.

Bidirectional Search

Bidirectional search combines forward selection and backward elimination to iteratively add and remove features from the feature subset. The algorithm starts with an empty set of features and adds features one at a time using forward selection. After adding each feature, the algorithm evaluates the performance of the current feature subset and removes the least important feature using backward elimination if necessary. The process continues until a stopping criterion is met, such as reaching a specified number of features or no improvement in performance. Bidirectional search aims to strike a balance between the efficiency of forward selection and the effectiveness of backward elimination.

Embedded Methods

Embedded methods integrate feature selection directly into the model training process, where feature importance is learned as part of the model construction. These methods are typically specific to certain learning algorithms that inherently perform feature selection during training, Mahendran, N., & PM, D. R. V. (2022), Chen, C. W., Tsai, Y. H., Chang, F. R., & Lin, W. C. (2020).

Lasso (L1 Regularization)

Lasso, short for Least Absolute Shrinkage and Selection Operator, is a linear regression technique that penalizes the absolute size of coefficients in the regression model. The L1 regularization term added to the standard linear regression objective function encourages sparse solutions by forcing some coefficients to be exactly zero. As a result, Lasso inherently performs feature selection by shrinking the coefficients of irrelevant features to zero, effectively removing them from the model. The strength of regularization is controlled by a hyperparameter (lambda or alpha), which determines the trade-off between model simplicity and accuracy. Lasso is particularly useful when dealing with high-dimensional datasets where feature selection is crucial for model interpretability and generalization.

Decision Trees

Decision trees are non-parametric models that recursively partition the feature space into regions based on feature values. During the construction of a decision tree, features are selected at each node to split the data into subsets that are as pure (homogeneous with respect to the target variable) as possible. Features that lead to the most significant reduction in impurity (e.g., Gini impurity or entropy) are selected for splitting, implicitly ranking the importance of features. Once the decision tree is trained, feature importance can be inferred from the frequency with which a feature is used for splitting across all nodes in the tree. Decision tree-based models, such as Random Forests and Gradient Boosted Trees, extend this concept to ensemble learning, where multiple decision trees are combined to further improve feature selection and model performance.

Random Forest Feature Importance

Random Forest is an ensemble learning technique that constructs multiple decision trees using bootstrapped samples of the dataset and random subsets of features. The importance of a feature in a Random Forest model is assessed based on the decrease in node impurity (e.g., Gini impurity or entropy) caused by splitting on that feature across all trees in the forest. Features that result in greater reductions in impurity are deemed more important and are assigned higher feature importance scores. Random Forest feature importance provides a robust and interpretable measure of feature relevance, taking into account interactions among features and their collective impact on model performance.

Elastic Net

Elastic Net is a linear regression technique that combines L1 (Lasso) and L2 (Ridge) regularization penalties to overcome some limitations of Lasso, such as instability and inconsistency in variable selection. Similar to Lasso, Elastic Net encourages sparsity in the solution by penalizing the absolute size of coefficients (L1 penalty). Additionally, Elastic Net includes an L2 penalty term that penalizes the square of the coefficients, which helps to handle multicollinearity and stabilize the selection of correlated features. By tuning the mixing parameter between L1 and L2 penalties, Elastic Net allows for flexible control over the trade-off between sparsity and model accuracy.

Gravitational Search Optimization (Gso)

Gravitational Search Optimization (GSO) is a metaheuristic optimization algorithm inspired by the laws of gravity and motion in physics. It simulates the interactions between masses (representing candidate solutions) within a search space, where masses exert gravitational forces on each other to guide the search process towards promising regions, Guha, R., Ghosh, M., Chakrabarti, A., Sarkar, R., & Mirjalili, S. (2020), Kumar, S., & John, B. (2021) [11] [12].

Step 1

Initialize a population of N masses (candidate solutions) randomly within the search space. Each mass represents a potential solution to the optimization problem.

Step 2

Define an objective function $f(x)$ that evaluates the fitness of each candidate solution x based on the optimization task at hand. The objective function maps each solution x to a real-valued fitness score representing its quality.

Step 3

- *Gravitational Force Calculation*

The gravitational force F_i acting on each mass m_i is calculation based on the Newtonian law of gravity, considering the interaction between the mass m_i and all other masses in the population. The gravitational force F_i acting on mass m_i is given by:

$$F_i = G.\left(\frac{m_i m_j}{r_{ij}^2}\right).\hat{r}_{ij} \quad (3.1)$$

G is the gravitational constant. m_i *and* m_j are the masses of masses i and j respectively. r_{ij} is the distance between masses i and j. $\hat{r}_{ij}$ is the unit vector pointing from mass i to mass j.

Step 4

- *Acceleration Calculation*

The acceleration a_i experienced by each mass m_i is computed by dividing the gravitational force acting on it by its mass m_i:

$$a_i = \frac{F_i}{m_i} \quad (3.2)$$

Step 5

Velocity and Position Update: The velocity v_i and position x_i of each mass m_i are updated based on its current velocity, acceleration, and position:

$$v_i = v_i + a_i.\Delta t \quad (3.3)$$

$$x_i = x_i + v_i.\Delta t \quad (3.4)$$

Where Δt is the time step. The updates are performed iteratively until a termination criterion is met, such as a maximum number of iterations or convergence.

Step 6

- *Solution Evaluation and Selection*

Evaluate the fitness of each updated solution x_i using the objective function $f(x)$. Select the best solutions based on their fitness scores for the next iteration.

Step 7

- *Termination*

Repeat steps 3 to 6 until a termination criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory solution quality.

Procedure for GSO based Feature Selection Method

Procedure GSO_Feature_Selection(dataset, population_size, max_iterations):

Initialize:

- Randomly generate an initial population of candidate feature subsets (masses).
- Set gravitational constant G.
- Set parameters for acceleration calculation, such as time step Δt.
- Set termination criterion (e.g., maximum number of iterations).

Repeat for each iteration until termination criterion is met:

For each mass in the population:

Compute fitness value of the feature subset using a classification algorithm:

- Train a classifier (e.g., GBT, RF) using the features in the subset.
- Evaluate the classifier's performance (e.g., accuracy, F1-score) on the dataset.
- Assign the fitness value to the mass based on the classifier's performance.

Calculate gravitational forces and accelerations:

For each mass in the population:

For each other mass in the population:

Compute distance between masses.

Compute gravitational force between masses using Newton's law of gravity.

Compute acceleration experienced by each mass.

Update velocities and positions of masses:

For each mass in the population:

Update velocity using acceleration and time step.

Update position using velocity and time step.

Evaluate and select new solutions:

For each updated mass:

Compute fitness value of the feature subset using the classifier.

Select best solutions:

Select the best-performing feature subsets based on their fitness values.

Termination:

If termination criterion is met (e.g., maximum number of iterations), stop the algorithm.

Return the best-performing feature subset found during the optimization process.

Particle Swarm Optimization (Pso)

Particle Swarm Optimization (PSO) can be adapted for feature selection by representing each particle as a binary vector indicating the presence or absence of features. The PSO algorithm then optimizes this binary vector to find the optimal subset of features that maximizes a predefined objective function, Kılıç, F., Kaya, Y., & Yildirim, S. (2021), Hu, Y., Zhang, Y., & Gong, D. (2020) [13] [14].

Step 1

- *Initialization*

Initialize a population of particles, each representing a potential feature subset, randomly within the search space. Each particle is represented by a binary vector x_i of size D, where D is the number of features in the dataset. Each element x_{ij} is the binary vector indicates the presence (1) or absence (0) of the j-th feature in the subset represented by particle i.

Step 2

- *Objective Function*

Define an objective function $f(x)$ that evaluates the fitness of each particle's feature subset based on the optimization task at hand. The objective function maps each solution x to a real valued fitness score representing its quality. For feature selection, the objective function typically measures the performance of a classification algorithm trained on the selected feature subset. Common performance metrics include accuracy, F1-score, or area under the ROC curve (AUC).

Step 3

- *Velocity Update*

The velocity v_i of each particle i is updated based on its current velocity, its best previous position p_i, and the global best position p^* found by any particle in the swarm:

$$v_i(t+1) = w.v_i(t) + c_i.r_i \odot (p_i(t) - x_i(t)) + c_2.r_2 \odot (p^*(t) - x_i(t)) \tag{3.5}$$

Where w is the inertia weight that controls the impact of the previous velocity. c_i and c_2 are acceleration coefficients representing cognitive and social components, respectively. r_i and r_2 are random vectors sampled uniformly from the range [0,1]. $\odot$ denotes element-wise multiplication.

Step 4

- *Position Update*

The position x_i of each particle i is updated based on its current position and velocity:

$$x_i(t+1) = clip(x_i(t) + v_i(t+1)) \tag{3.6}$$

Where $clip(.)$ ensures that the binary vector x_i remains within the valid range [0,1].

Step 5

- *Global Best Update*

Update the global best position p^* found by any particle in the swarm:

$$p^*(t+1) = arg\min_{p_i} f(p_i(t+1)) \tag{3.7}$$

Step 6

- *Termination*

Repeat steps 3 to 5 until a termination criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory solution quality.

Procedure for PSO based Feature Selection Method

Procedure PSO_Feature_Selection(dataset, population_size, max_iterations):

Initialize:

- Randomly generate an initial population of particles, each representing a potential feature subset.
- Set parameters for PSO, including inertia weight (w), cognitive coefficient (c1), and social coefficient (c2).
- Set termination criterion (e.g., maximum number of iterations).

Repeat for each iteration until termination criterion is met:

For each particle in the population:

Evaluate the fitness of the feature subset represented by the particle:

- Train a classifier (e.g., SVM, Decision Trees) using the features in the subset.
- Evaluate the classifier's performance (e.g., accuracy, F1-score) on the dataset.
- Assign the fitness value to the particle based on the classifier's performance.

Update particle velocities and positions:

For each particle in the population:

Update velocity using previous velocity, cognitive and social components:

v_i(t+1) = w * v_i(t) + c1 * r1 * (p_i(t) - x_i(t)) + c2 * r2 * (p_best(t) - x_i(t))

Update position using velocity:

x_i(t+1) = x_i(t) + v_i(t+1)

Update personal best positions:

For each particle in the population:

If the fitness of the current position is better than the personal best fitness:

Update personal best position to the current position.

Update global best position:

Determine the particle with the best fitness in the entire population.

Update global best position to the personal best position of that particle.

Termination:

If termination criterion is met (e.g., maximum number of iterations), stop the algorithm.

Return the best-performing feature subset found during the optimization process.

Proposed Metaheuristic Fusion of Gravitational Search and Particle Swarm Optimization (GSPSO) Based Feature Selection Method

The proposed metaheuristic fusion of Gravitational Search Optimization (GSO) and Particle Swarm Optimization (PSO), known as GSPSO, for feature selection in the healthcare domain aims to improve the accuracy and efficiency of healthcare decision support systems. This method leverages the complementary strengths of both GSO and PSO to identify optimal subsets of features from medical datasets, which can subsequently be used for tasks such as disease diagnosis, patient risk prediction, and treatment recommendation. Healthcare datasets often contain a large number of features, including patient demographics, clinical variables, and medical test results. Selecting relevant features is crucial for building accurate predictive models and decision support systems in healthcare. GSPSO offers a powerful approach to efficiently search through the vast feature space and identify subsets of features that are most informative for healthcare-related tasks.

GSO is known for its ability to efficiently explore the search space and identify promising regions using gravitational forces. PSO excels at exploiting known good solutions and fine-tuning the search process based on particle velocities and positions. The fusion of GSO and PSO in GSPSO leverages the exploration capabilities of GSO and the exploitation capabilities of PSO to enhance the feature selection process.

Each particle in the GSPSO algorithm represents a potential feature subset, encoded as a binary vector indicating the presence or absence of each feature. The objective function evaluates the fitness of each particle's feature subset based on its performance in healthcare-related tasks, such as disease classification or patient outcome prediction. Gravitational forces from GSO influence the particle velocities in PSO, guiding particles towards promising regions of the feature space. Particle velocities and positions are updated iteratively based on the gravitational forces and the cognitive and social components in PSO.

Step 1: Initialization

Initialize a population of particles, each representing a potential feature subset, randomly within the search space. Each particle is represented by a binary vector x_i of size D, where D is the number of features in the dataset. Each element $x_{i,j}$ of the binary vector indicates the presence (1) or absence (0) of the j-th feature in the subset represented by particle i.

Step 2: Objective Function

Define an objective function $f(x)$ that evaluates the fitness of each particle's feature subset based on the optimization task at hand. The objective function maps each solution X to a real-valued fitness score representing its quality. For feature selection, the objective function typically measures the performance of a classification algorithm trained on the selected feature subset. Common performance metrics include accuracy, F1-score, or area under the ROC curve (AUC).

Step 3: Velocity Update (using a fusion of GSO and PSO)

Compute the gravitational force F_i acting on each particle i using GSO equations (3.1). Compute the velocity v_i of each particle i using PSO equations, incorporating the gravitational force:

$$v_i(t+1) = w.v_i(t) + c_i.r_i \odot (p_i(t) - x_i(t)) + c_2.r_2 \odot (p^*(t) - x_i(t)) + \frac{F_i}{m_i}$$

Where F_i is the gravitational force acting on particle i calculated using equation (3.1). m_i is the mass of the particle i, which can be set to a constant or dynamically adjusted.

Step 4: Position Update

The position x_i of each particle i is updated based on its current position and velocity using the equation (3.6).

Step 5: Global Best Update

Update the global best position p^* using the equation (3.7) found by any particle in the swarm

Step 6: Termination

Repeat steps 3 to 5 until a termination criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory solution quality.

The GSPSO-based feature selection method combines the exploration capabilities of GSO with the exploitation capabilities of PSO to efficiently explore the search space and converge towards promising feature subsets. By incorporating gravitational forces into the velocity update equation of PSO, GSPSO enhances the search process and improves the quality of selected feature subsets. Parameter tuning, including the inertia weight *w*, cognitive coefficient c_i, and social coefficient c_2, is crucial for the performance of GSPSO-based feature selection. Additionally, termination criteria and initialization strategies are important considerations for applying GSPSO to feature selection tasks.

Procedure for Proposed GSPSO based Feature Selection Method

Procedure GSPSO_Feature_Selection(dataset, population_size, max_iterations):

Initialize:

- Randomly generate an initial population of particles, each representing a potential feature subset, within the search space.
- Set parameters for GSO (gravitational constant, mass, gravitational force calculation).
- Set parameters for PSO (inertia weight, cognitive and social coefficients).
- Set termination criterion (e.g., maximum number of iterations).

Repeat for each iteration until termination criterion is met:

For each particle in the population:

Evaluate the fitness of the feature subset represented by the particle:

- Train a classifier (e.g., GBN, RF, SVM) using the features in the subset.
- Evaluate the classifier's performance (e.g., accuracy, F1-score) on the dataset.
- Assign the fitness value to the particle based on the classifier's performance.

Calculate gravitational forces:

For each particle in the population:

Compute gravitational forces acting on the particle based on GSO equations.

Update particle velocities and positions using PSO:

For each particle in the population:

Update velocity using PSO equations, incorporating gravitational forces:

Compute velocity update based on PSO equations with gravitational force component.

Update position using velocity:

Update position based on the calculated velocity, ensuring it remains within the valid range [0,1].

Update personal best positions:

For each particle in the population:

If the fitness of the current position is better than the personal best fitness:

Update personal best position to the current position.

Update global best position:

Determine the particle with the best fitness in the entire population.

Update global best position to the personal best position of that particle.

Termination:

If termination criterion is met (e.g., maximum number of iterations), stop the algorithm.

Return the best-performing feature subset found during the optimization process.

Result And Discussions

Performance Metrics

Table 1 gives the performance metrics used in this research work to evaluate the performance of the proposed GSPSO feature selection method.

Table 2 depicts the Classification Accuracy (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 1 gives the graphical

Table 1: Performance Metrics

Metrics	*Equation*
Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
True Positive Rate (TPR)	$\frac{TP}{TP+FN}$
False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Precision	$\frac{TP}{TP+FP}$
Specificity	1-FPR
Miss Rate	1- TPR
False Discovery Rate	1- Precision

Table 2: Classification Accuracy (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*Classification Accuracy (in %) by Classification Techniques*		
	GBT	*RF*	*SVM*
Original Dataset	55.38	45.63	43.32
Proposed GSPSO method	95.78	92.29	89.63
GSA	73.85	63.81	58.45
PSO	74.99	71.86	68.02
ABC	69.74	65.95	63.54
GA	68.68	62.65	61.45

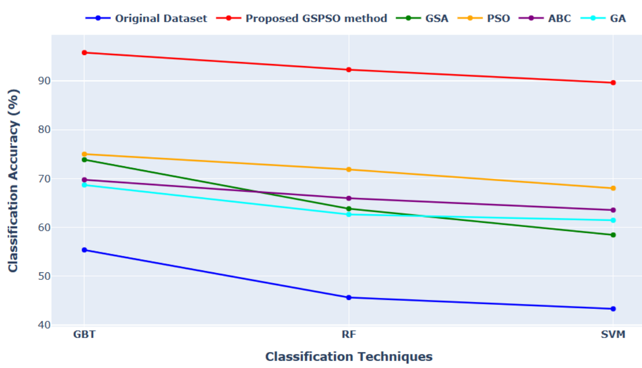

Figure 1: Graphical representation of the Classification Accuracy (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

representation of the Classification Accuracy (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. From the Table 2 and Figure 1, it is clear that the Proposed GSPSO with GBT gives better accuracy than the existing feature selection methods.

Table 3 depicts the True Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 2 gives the graphical representation of the True Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. From the Table 3 and Figure 2, it is clear that the Proposed GSPSO with GBT gives better true positive rate than the existing feature selection methods.

Table 4 depicts the False Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 3 gives the graphical representation of the False Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets

Table 3: True Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*True Positive Rate (in %) by Classification Techniques*		
	GBT	*RF*	*SVM*
Original dataset	54.49	44.54	42.23
Proposed GSPSO method	95.59	91.38	89.72
GSA	75.81	72.95	69.13
PSO	74.96	64.92	59.56
ABC	68.86	64.86	62.63
GA	67.77	61.56	60.53

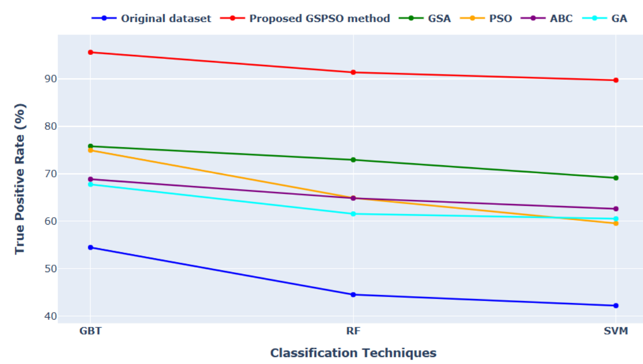

Figure 2: Graphical representation of the True Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Table 4: False Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*False Positive Rate (in %) by Classification Techniques*		
	GBT	*RF*	*SVM*
Original dataset	53.61	64.17	65.69
Proposed GSPSO method	5.94	6.41	9.54
GSA	22.42	30.18	33.47
PSO	27.53	33.62	34.47
ABC	38.82	44.51	45.84
GA	41.72	47.34	48.73

using GBT, RF and SVM. From the Table 4 and Figure 3, it is clear that the Proposed GSPSO with GBT gives reduced FPR than the existing feature selection methods.

Table 5 depicts the Precision (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 4 gives the graphical representation of the Precision (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC

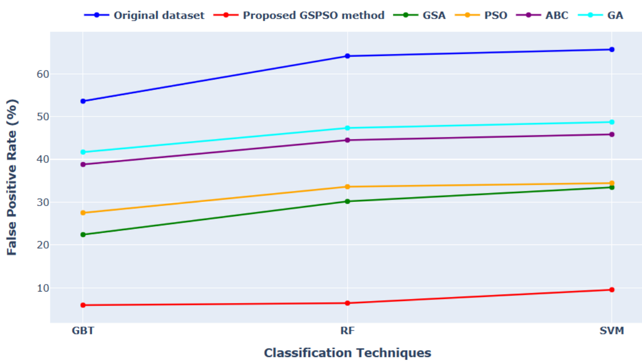

Figure 3: Graphical representation of the False Positive Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

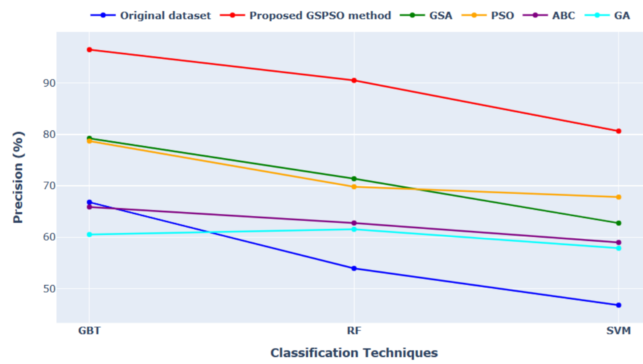

Figure 4: Graphical representation of the Precision (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Table 5: Precision (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*Precision (in %) by Classification Techniques*		
	GBT	*RF*	*SVM*
Original dataset	66.81	53.92	46.76
Proposed GSPSO method	96.52	90.53	80.66
GSA	79.25	71.38	62.74
PSO	78.72	69.82	67.81
ABC	65.88	62.76	58.97
GA	60.52	61.53	57.85

Table 6: Specificity (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*Specificity (in %) by Classification Techniques*		
	GBT	*RF*	*SVM*
Original dataset	46.39	35.83	34.31
Proposed GSPSO method	94.06	93.59	90.46
GSA	77.58	69.82	66.53
PSO	72.47	66.38	65.53
ABC	61.18	55.49	54.16
GA	58.28	52.66	51.27

and PSO method processed datasets using GBT, RF and SVM. From the Table 5 and Figure 4, it is clear that the Proposed GSPSO with GBT gives increased precision than the existing feature selection methods.

Table 6 depicts the Specificity (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 5 gives the graphical representation of the Specificity (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. From the Table 6 and Figure 5, it is clear that the Proposed GSPSO with GBT gives increased specificity than the existing feature selection methods.

Table 7 depicts the Miss Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 6 gives the graphical representation of the Miss Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. From the Table 7 and Figure 6, it is clear that the Proposed GSPSO with GBT gives reduced miss rate than the existing feature selection methods.

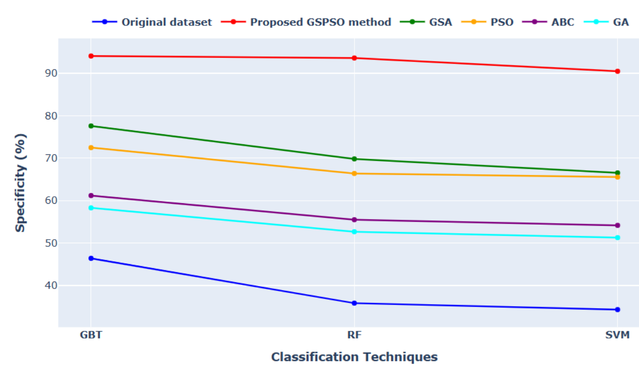

Figure 5: Graphical representation of the Specificity (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Table 8 depicts the False Discovery Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM. Figure 7 gives the graphical representation of the False Discovery Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets

Table 7: Miss Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*Classification Techniques*		
	GBT	*RF*	*SVM*
Original dataset	45.51	55.46	57.77
Proposed GSPSO method	4.41	8.62	10.28
GSA	24.19	27.05	30.87
PSO	25.04	35.08	40.44
ABC	31.14	35.14	37.37
GA	32.23	38.44	39.47

Table 8: False Discovery Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

Feature Selection Methods	*Classification Techniques*		
	GBT	*RF*	*SVM*
Original dataset	33.19	46.08	53.24
Proposed GSPSO method	3.48	9.47	19.34
GSA	20.75	28.62	37.26
PSO	21.28	30.18	32.19
ABC	34.12	37.24	41.03
GA	39.48	38.47	42.15

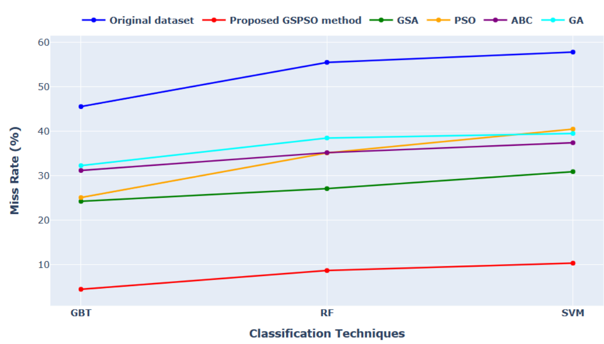

Figure 6: Graphical representation of the Miss Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

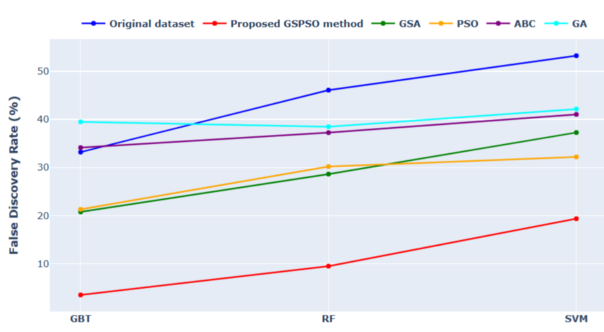

Figure 7: Graphical representation of the False Discovery Rate (in %) obtained for the HR dataset using original dataset, Proposed GSPSO method, GSA, GA, ABC and PSO method processed datasets using GBT, RF and SVM

using GBT, RF and SVM. From the Table 8 and Figure 7, it is clear that the Proposed GSPSO with GBT gives reduced FDR than the existing feature selection methods.

Conclusion

This study presents a novel approach integrating Particle Swarm Optimization (PSO) and Glowworm Swarm Optimization (GSO) for enhanced HR analytics, focusing on achieving high accuracy in employee classification and predictive modeling tasks. The combined PSO-GSO methodology is designed to optimize feature selection and parameter tuning, improving the model's performance by enhancing its predictive and classification accuracy across various HR-related datasets. By leveraging the strengths of both PSO's global exploration capabilities and GSO's localized search efficiency, this hybrid approach achieves superior convergence and avoids common pitfalls, such as local minima, in optimization.

Performance metrics such as accuracy, precision, recall, false positive rate (FPR), miss rate, false discovery rate (FDR), and true positive rate (TPR) were used to evaluate the efficacy of the proposed PSO-GSO model. The results indicate that the model provides notable improvements, especially in minimizing error rates like the FPR and miss rate while maximizing the true positive rate. This translates into higher precision and recall, ensuring that the model accurately identifies relevant cases (e.g., potential employee turnover, performance risks, etc.) and reduces the likelihood of false alarms. Additionally, the low false discovery rate underscores the reliability of predictions, thus enhancing the credibility of the model in real-world HR applications.

The PSO-GSO approach's balanced performance across these metrics suggests its utility in HR analytics for applications requiring both high accuracy and reliability. The model's effectiveness in maintaining high accuracy, precision, and recall while minimizing error rates offers significant value for organizations aiming to adopt data-driven HR strategies. Future work could expand on this framework by incorporating additional optimization algorithms and exploring hybrid approaches to further enhance predictive capabilities and interpretability in HR analytics applications. Ultimately, the proposed PSO-GSO model demonstrates a robust approach to optimizing HR analytics, empowering HR departments with actionable insights for strategic workforce management.

References

Chen, C. W., Tsai, Y. H., Chang, F. R., & Lin, W. C. (2020). Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. Expert Systems, 37(5), e12553.

Dash, M., & Liu, H. (1997). Feature selection for classification. Intelligent data analysis, 1(1-4), 131-156.

Durairaj, M., & Poornappriya, T. S. (2020). Why feature selection in data mining is prominent? A survey. In Proceedings of International Conference on Artificial Intelligence, Smart Grid and Smart City Applications: AISGSC 2019 (pp. 949-963). Springer International Publishing.

Ghosh, M., Guha, R., Sarkar, R., & Abraham, A. (2020). A wrapper-filter feature selection technique based on ant colony optimization. Neural Computing and Applications, 32, 7839-7857.

Guha, R., Ghosh, M., Chakrabarti, A., Sarkar, R., & Mirjalili, S. (2020). Introducing clustering based population in binary gravitational search algorithm for feature selection. Applied Soft Computing, 93, 106341.

Hu, Y., Zhang, Y., & Gong, D. (2020). Multiobjective particle swarm optimization for feature selection with fuzzy cost. IEEE Transactions on Cybernetics, 51(2), 874-888.

Jha, K., & Saha, S. (2021). Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique. Applied Soft Computing, 98, 106823.

Kılıç, F., Kaya, Y., & Yildirim, S. (2021). A novel multi population based particle swarm optimization for feature selection. Knowledge-Based Systems, 219, 106894.

Kumar, S., & John, B. (2021). A novel gaussian based particle swarm optimization gravitational search algorithm for feature selection and classification. Neural Computing and Applications, 33(19), 12301-12315.

Le, T. M., Vo, T. M., Pham, T. N., & Dao, S. V. T. (2020). A novel wrapper–based feature selection for early diabetes prediction enhanced with a metaheuristic. IEEE access, 9, 7869-7884.

Mahendran, N., & PM, D. R. V. (2022). A deep learning framework with an embedded-based feature selection approach for the early detection of the Alzheimer's disease. Computers in Biology and Medicine, 141, 105056.

Navot, A. (2006). On the role of feature selection in machine learning (Doctoral dissertation, Hebrew University).

Nourmohammadi-Khiarak, J., Feizi-Derakhshi, M. R., Behrouzi, K., Mazaheri, S., Zamani-Harghalani, Y., & Tayebi, R. M. (2020). New hybrid method for heart disease diagnosis utilizing optimization algorithm in feature selection. Health and Technology, 10(3), 667-678.

Siddiqi, M. A., & Pak, W. (2020). Optimizing filter-based feature selection method flow for intrusion detection system. Electronics, 9(12), 2114.