Stability-Weighted Feature Selection with Adaptive PSO-SGD for Neural Network-Based Predictive Hiring

Yassine Temsamani Khallouk, Said Achchab
Mohammed V University in Rabat, Morocco

***Abstract*—Predictive hiring leverages machine learning to forecast candidate success, yet existing approaches suffer from two limitations: reliance on single-method feature selection that lacks robustness, and sensitivity to neural network initialization that impairs convergence. This study introduces two contributions integrated into a unified framework. First, Stability-Weighted Multi-Criteria Feature Selection (SW-MCFS) is proposed, which aggregates four heterogeneous scoring methods—Mutual Information, Wald statistical significance, Fisher discriminant loading, and Permutation Importance—through a cross-validation stability-weighted consensus function. Unlike single-method approaches, SW-MCFS weights each method proportionally to its ranking consistency across folds, producing robust and data-driven feature subsets. Second, Adaptive Particle Swarm Optimization (APSO) is introduced, a PSO variant featuring fitness-landscape-aware inertia adaptation and Lévy flight perturbation for stagnation escape. The framework is evaluated on $10,247$ recruitment records from a North African telecommunications company and benchmarked against Random Forest, XGBoost, SVM, standard ANNs, and classical LR/DA-based approaches through 10-fold cross-validation. The integrated SW-MCFS-APSO-SGD framework achieves 76.8% accuracy, significantly outperforming XGBoost (73.8%, $p = 0.012$), standard PSO-SGD (75.2%, $p = 0.041$), and LR-based feature selection (74.6%, $p = 0.028$). Ablation studies confirm that SW-MCFS contributes 1.6% accuracy gain over single-method selection, while APSO improves performance by 0.8% with 31% faster convergence compared to standard PSO. SHAP analysis reveals communication skills, experience, and seniority as dominant predictors with minimal demographic influence. It is noted that the accuracy ceiling may partly reflect inherent label noise in subjective performance assessments. The proposed framework demonstrates effectiveness on organizational recruitment data, warranting further cross-domain validation to establish broader generalizability.**

***Keywords*—Predictive hiring; Multi-Criteria Feature Selection; Adaptive Particle Swarm Optimization; stability-weighted consensus; neural networks; HR analytics; ensemble methods**

I. Introduction

A. Background and Motivation

Modern recruitment processes generate large volumes of structured and unstructured data, including resumes, assessments, interviews, and work history. Traditional hiring practices struggle to analyze this information objectively and at scale, resulting in inefficiencies, bias, and suboptimal talent selection [1], [2], [30]. Predictive hiring applies machine learning to forecast candidate success by leveraging historical employment data to identify performance-related patterns [3], [32].

Recent advances in artificial intelligence (AI) have enabled automated candidate evaluation through natural language processing (NLP) for resume analysis [4], psychometric profiling [5], and automated interview assessment [6].

B. AI-Driven Transformation of Human Resource Information Systems

AI integration transforms Human Resource Information Systems (HRIS) from transactional platforms into intelligent decision-support systems. Traditional HRIS mainly manages payroll, personnel records, and compliance [7]. With AI integration, these systems evolve into advanced analytics-driven infrastructures [8], [9].

AI technologies, including ML, deep learning (DL), NLP, and large language models (LLM), enable HRIS to process heterogeneous and large-scale workforce data. This supports evidence-based human resource management and strengthens core functions including talent acquisition, workforce planning, and performance management [10]. Strategically, AI enables HR systems to transition toward intelligent and data-driven workforce management with enhanced analytical capabilities.

C. AI Techniques for Recruitment Automation and Predictive Modeling

NLP techniques such as tokenization, semantic embedding, and named entity recognition enable automated extraction of structured features from textual recruitment data [11], [12]. Transformer-based architectures [42] improve semantic similarity estimation and candidate–job matching through contextual representation learning.

LLM further enhances automation by improving feature representation and supporting intelligent screening and classification [13]. Predictive hiring systems then use these extracted features to estimate post-hire performance using ML models. This study integrates advanced feature selection and optimization methods to improve predictive robustness within AI-driven HR environments.

D. Research Challenges

Despite recent advances, two fundamental challenges remain:

- *Feature Selection Fragility:* Most existing studies rely on a single feature selection technique such as correlation filtering, LR significance testing, or tree-based

importance ranking. These approaches often produce unstable feature subsets that vary across data partitions and model assumptions [14]. However, no consensus-based framework systematically aggregates multiple selection criteria while accounting for stability across cross-validation folds.

- *Optimization Sensitivity:* Standard Particle Swarm Optimization (PSO) employs fixed or predefined inertia weight schedules that do not adapt to the fitness landscape topology [15]. When used to optimize neural network (NN) parameters, this limitation may result in premature convergence or insufficient exploration, reducing predictive performance and generalization ability.

E. Research Contributions

To address these challenges, this study makes the following contributions:

1) Stability-Weighted Multi-Criteria Feature Selection (SW-MCFS): A feature selection framework is proposed that aggregates multiple heterogeneous scoring methods, including (1) Mutual Information (MI), (2) Wald Statistical Significance (WSS), (3) Fisher Discriminant Loading (FDL), and (4) Permutation Importance (PI), into a unified consensus ranking. Each method is weighted based on its cross-validation stability to enhance robustness. To the best of the authors' knowledge, this is the first stability-weighted multi-criteria consensus framework applied to predictive hiring feature selection.

2) Adaptive PSO (APSO): An adaptive PSO variant is introduced incorporating (1) fitness-based inertia adaptation, (2) swarm diversity monitoring, and (3) Lévy flight perturbations to enhance exploration and prevent premature convergence during NN weight optimization.

3) Integrated hybrid learning framework: An end-to-end pipeline is developed that integrates SW-MCFS with adaptive APSO for initialization and stochastic gradient descent (SGD) for training. This joint design improves feature robustness and model optimization within a unified framework that has not been explored in prior predictive hiring studies.

4) Rigorous empirical validation: Comprehensive evaluation on $10,247$ records using 10-fold cross-validation, ablation analysis, statistical tests, and baseline comparisons to demonstrate robustness and effectiveness.

F. Paper Organization

The remainder of this study is organized as follows: Section II reviews related work and theoretical foundations. Section III describes the proposed optimization framework. Section IV presents the experimental setup. Section V and Section VI reports the results and discussion. Finally, Section VII concludes the study and outlines future research directions.

II. Related Work

A. Feature Selection in HR Analytics

Feature selection in HR prediction typically include filter methods (e.g., correlation, MI), wrapper methods (e.g., recursive feature elimination), and embedded methods (e.g., L1 regularization, tree-based importance) [16]. Most studies rely on a single technique such as Logistic Regression (LR) significance testing [17] or Random Forest (RF) importance [18].

However, these approaches often suffer from instability and bias. Filter methods ignore feature interactions, wrapper methods are computationally expensive, and embedded methods depend on the base model [19]. Although ensemble feature selection has been explored in other domains [20], stability-aware consensus strategies remain limited in HR prediction. The proposed SW-MCFS framework addresses this gap by incorporating stability-weighted aggregation.

B. PSO Variants for NN Optimization

PSO is widely used for NN optimization due to its global search ability [15]. Standard PSO uses fixed inertia strategies, which limit adaptability to complex landscapes.

Adaptive variants improve exploration through dynamic inertia control [21], state-based adaptation [22], or Lévy flight perturbations [23]. However, existing methods rarely combine fitness-aware adaptation with Lévy-based exploration for predictive hiring models. The proposed APSO variant integrates these mechanisms to improve optimization stability.

C. AI in Predictive Hiring

ML for recruitment has evolved from traditional classifiers to deep learning models. NN-based person–job fit models achieve competitive accuracy [24], [33], while co-attention architectures and ensemble methods, including gradient boosting variants such as XGBoost [26] and CatBoost [41] further improve performance [25].

Despite these advances, feature selection and model training are usually treated separately. Integrated optimization within a unified pipeline remains underexplored.

D. Research Gaps

Current literature lacks: 1) stability-weighted multi-criteria feature selection for HR prediction, 2) adaptive swarm optimization combining fitness-aware control and Lévy perturbation for neural training, and 3) unified frameworks that jointly optimize feature robustness and model learning. These gaps motivate the proposed SW-MCFS-APSO-SGD framework.

III. Proposed SW-MCFS–APSO Hybrid Optimization Framework

This section presents the proposed SW-MCFS–APSO–SGD framework, depicted in Fig. 1, that integrates stability-aware feature selection, adaptive optimization, and gradient-based training for robust predictive modeling.

A. Problem Formulation and Cross-Validation Framework

Let the training dataset be defined as Eq. (1):

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}, \tag{1}$$

where,

- N denotes the number of recruitment records,

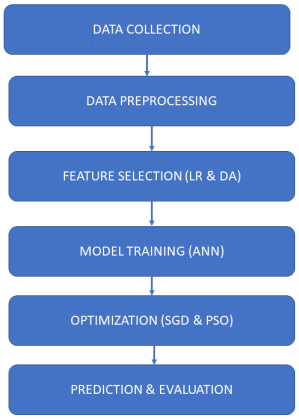

Fig. 1. SW-MCFS-APSO-SGD framework. Phase 1: Multi-criteria feature selection with stability weighting. Phase 2: APSO global search with FLAI, LFP, and EGR. Phase 3: SGD fine-tuning with early stopping.

- $\mathbf{x}_i = \{x_{i,j}\}_{1\le j\le d} \in \mathbb{R}^d$ is the raw feature vector of candidate i $(i \le N)$,
- d is the number of original features,
- $y_i \in \{0, 1\}$ is the binary job performance label, with $i \le N$.

Accordingly, the feature matrix can be expressed as Eq. (2):

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} \in \mathbb{R}^{N\times d}. \tag{2}$$

Each row corresponds to a candidate, and each column corresponds to a feature. To evaluate feature stability and prevent data leakage, K-fold cross-validation (CV) is adopted on $\mathcal{D}$. For each $k \le K$, the dataset $\mathcal{D}$ is partitioned into training and validation set as Eq. (3):

$$\mathcal{D} = \mathcal{D}_{\text{tr}}^{(k)} \cup \mathcal{D}_{\text{val}}^{(k)}, \quad 1 \le k \le K, \tag{3}$$

such that for each fold k [see Eq. (4)]:

$$\begin{cases} \mathcal{D}_{\text{tr}}^{(k)} \cap \mathcal{D}_{\text{val}}^{(k)} = \emptyset \\ \left|\mathcal{D}_{\text{val}}^{(k)}\right| = \lfloor \frac{N}{K} \rfloor, \quad \left|\mathcal{D}_{\text{tr}}^{(k)}\right| = N_k \triangleq N - \lfloor \frac{N}{K} \rfloor \end{cases}. \tag{4}$$

B. Multi-Criteria Feature Selection

Initially, the proposed SW-MCFS algorithm is applied to reduce the original feature set to a selected subset $\mathcal{F}$, where $d^* = |\mathcal{F}|$ denotes the number of selected features satisfying $d^* \le d$.

The reduced feature matrix becomes [see Eq. (5)]:

$$\mathbf{M}_{\mathcal{F}} \in \mathbb{R}^{N\times d^*}. \tag{5}$$

To achieve a robust and stable ranking, multiple complementary feature relevance criteria are aggregated. In particular, **four** scoring methods are employed to evaluate feature importance from statistical, information-theoretic, and model-driven perspectives.

1) Mutual information: For each feature $x_{i,j}$ (the j-th feature of sample i), the MI quantifies its dependency with the binary label y_i by measuring the reduction in uncertainty of y_i given $x_{i,j}$.

Since features are continuous, the joint and marginal distributions are estimated using kernel density estimation or discretization. The MI score for feature j is computed in fold k, as in [27] [see Eq. (6)]:

$$\mathcal{M}_j^{(k)} = \sum_{x,y} p_{X_j,Y}^{(k)}(x,y) \log \frac{p_{X_j,Y}^{(k)}(x,y)}{p_{X_j}^{(k)}(x)\, p_Y^{(k)}(y)}, j \le d, k \le K \tag{6}$$

where, X_j and Y denote the j-th feature and label RVs, respectively, and the joint distribution $p_{X_j,Y}^{(k)}(x,y)$ and the marginals $p_{X_j}^{(k)}(x)$ and $p_Y^{(k)}(y)$ are estimated empirically from $\mathcal{D}_{\text{tr}}^{(k)}$. Of note, a larger $\mathcal{M}_j^{(k)}$ implies stronger relevance of feature j to the target label.

For continuous features, $\mathcal{M}_j^{(k)}$ is approximated using the k-nearest-neighbor estimator [28]:

$$\widehat{\mathcal{M}}_j^{(k)} = \psi(k_{\text{nn}}) + \psi(N_k) - \frac{1}{N_k} \sum_{i=1}^{N_k} \Big[\psi(n_x^{(i)}+1) + \psi(n_y^{(i)}+1)\Big], \tag{7}$$

where, $n_x^{(i)}$ and $n_y^{(i)}$ denote the number of samples (excluding the i-th one) whose distances to sample i in the X_j and Y marginal spaces are smaller than the k_{nn}-nearest neighbor (NN) radius ε_i, which is determined by the distance to the k_{nn}-th NN in the joint space.

Lastly, to make scores comparable across features, the normalized MI score is defined based on the estimator in Eq. (7) as [see Eq. (8)]:

$$S_{j,k}^{(1,n)} = \frac{\widehat{\mathcal{M}}_j^{(k)}}{\max_{1\le \ell\le d} \widehat{\mathcal{M}}_\ell^{(k)}}, \quad 1 \le k \le K. \tag{8}$$

2) Wald statistical significance: For each fold, a univariate L2-regularized LR model is fitted to evaluate the statistical significance of feature X_j to the binary label Y. The conditional probability is modeled as Eq. (9):

$$\Pr(Y = 1 \mid X_j = x_{i,j}) = \frac{1}{1 + \exp\big(-\beta_{0,j}^{(k)} - \beta_{1,j}^{(k)} x_{i,j}\big)}. \tag{9}$$

The parameters $\beta_{0,j}^{(k)}$ and $\beta_{1,j}^{(k)}$ are estimated from $\mathcal{D}_{\text{tr}}^{(k)} \triangleq \{(x_{i,j}, y_i)\}_{i=1}^{N_k}$ via penalized maximum likelihood with L2 regularization to ensure numerical stability.

The null hypothesis $\mathcal{H}_0 : \beta_{1,j}^{(k)} = 0$ is tested using the Wald statistic [see Eq. (10)]:

$$W_j^{(k)} = \frac{(\hat{\beta}_{1,j}^{(k)})^2}{\widehat{\mathrm{Var}}(\hat{\beta}_{1,j}^{(k)})} \sim \chi^2(1), \qquad 1 \le j \le d, \tag{10}$$

where, $\chi^2(1)$ denotes the chi-square distribution with one degree of freedom, and $\widehat{\mathrm{Var}}(\hat{\beta}_{1,j}^{(k)})$ is obtained from the inverse Fisher information matrix.

The corresponding p-value is Eq. (11):

$$p_j^{(k)} = \Pr\big(\chi^2(1) \ge W_j^{(k)}\big) = 1 - F_{\chi^2(1)}(W_j^{(k)}), \tag{11}$$

where, $F_{\chi^2(1)}(\cdot)$ denotes the cumulative distribution function.

Similarly to Eq. (8), the normalized Wald-based score is defined as Eq. (12):

$$S_{j,k}^{(2,n)} = \frac{|\hat{\beta}_{1,j}^{(k)}|(1-p_j^{(k)})}{\max_{1\le\ell\le d}|\hat{\beta}_{1,\ell}^{(k)}|(1-p_\ell^{(k)})}. \tag{12}$$

This formulation jointly rewards large effect sizes and high statistical confidence while penalizing unstable estimates.

3) Fisher discriminant score: From Linear Discriminant Analysis [44], the discriminant direction that maximizes class separation in fold k is given by:

$$\mathbf{w}^{(*,k)} = (\mathbf{S}_W^{(k)})^{-1}(\boldsymbol{\mu}_1^{(k)} - \boldsymbol{\mu}_2^{(k)}), \tag{13}$$

where, $\mathbf{S}_W^{(k)}$ denotes the within-class scatter matrix computed on $\mathcal{D}_{\text{tr}}^{(k)}$ and $\boldsymbol{\mu}_c^{(k)}$ represents the mean vector of class c in fold k.

Each component $w_j^{(*,k)}$ of the discriminant direction in Eq. (13) measures the contribution of feature j to class separation. To obtain a scale-invariant measure, it is weighted by the pooled within-class standard deviation $\sigma_j^{(k)}$, and the normalized Fisher score in fold k is defined as Eq. (14):

$$S_{j,k}^{(3,n)} = \frac{|w_j^{(*,k)}|\,\sigma_j^{(k)}}{\max_{1\le\ell\le d}|w_\ell^{(*,k)}|\,\sigma_\ell^{(k)}}, \qquad 1 \le j \le d, \tag{14}$$

This normalization ensures comparability across features and highlights those with stronger discriminative power.

4) Permutation importance: Permutation importance is computed within each cross-validation fold using a baseline RF model with $B = 200$ trees [18], [29]. It quantifies the decrease in predictive performance when feature values are randomly permuted.

Let $f_\theta^{(k)}(\cdot)$ denote the classifier trained on the training subset $\mathcal{D}_{tr}^{(k)}$ in fold k. The accuracy functional evaluated on a dataset $\mathcal{D}$ in fold k is defined as Eq. (15):

$$\mathcal{A}^{(k)}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x}_k,y_k)\in\mathcal{D}} \mathbf{1}\big(f_\theta^{(k)}(\mathbf{x}_k) = y_k\big), \tag{15}$$

where, $\mathbf{1}(\cdot)$ is the indicator function.

The baseline accuracy in fold k is computed on the original validation set as Eq. (16):

$$\mathcal{A}_{\text{original}}^{(k)} = \mathcal{A}^{(k)}(\mathcal{D}_{\text{val}}^{(k)}). \tag{16}$$

To evaluate the importance of feature j in fold k, its values are randomly permuted across the validation samples, generating a modified dataset $\pi_r(\mathcal{D}_{\text{val}}^{(k)}, X_j)$. The permutation importance score is defined as the average reduction in accuracy [see Eq. (17)]:

$$\mathcal{P}_j^{(k)} = \mathcal{A}_{\text{original}}^{(k)} - \frac{1}{R}\sum_{r=1}^{R}\mathcal{A}^{(k)}\Big(\pi_r(\mathcal{D}_{\text{val}}^{(k)}, X_j)\Big); 1 \le j \le d, \tag{17}$$

where, R denotes the number of random permutations used in each fold.

Finally, the normalized importance score is given by Eq. (18):

$$S_{j,k}^{(4,n)} = \frac{\mathcal{P}_j^{(k)}}{\max_{1\le\ell\le d}\mathcal{P}_\ell^{(k)}}; 1 \le j \le d. \tag{18}$$

To further improve robustness, SW-MCFS assigns stability-driven weights to the four methods, as detailed in the subsequent subsection, thereby reducing the influence of methods exhibiting inconsistent rankings across folds.

C. Stability-Based Weighting

To quantify the reliability of each method, its ranking stability is measured as the average pairwise Spearman correlation between fold-specific rank vectors:

$$\rho_m = \frac{2}{K(K-1)}\sum_{k=1}^{K-1}\sum_{l=k+1}^{K}\rho_S\big(\mathbf{r}_m^{(k)}, \mathbf{r}_m^{(l)}\big), \tag{19}$$

where, Eq. (20):

$$\mathbf{r}_m^{(k)} = \big(r_{m,1}^{(k)}, \ldots, r_{m,d}^{(k)}\big), \quad m \le 4, 1 \le k \le K, \tag{20}$$

denote the rank vector of the d original features, and the Spearman rank correlation is:

$$\rho_S\big(\mathbf{r}_m^{(k)}, \mathbf{r}_m^{(l)}\big) = 1 - \frac{6\sum_{j=1}^{d}\big(r_{m,j}^{(k)} - r_{m,j}^{(l)}\big)^2}{d(d^2-1)}. \tag{21}$$

Substituting Eq. (21) into Eq. (19) yields Eq. (22):

$$\rho_m = 1 - \frac{12}{K(K-1)d(d^2-1)}\sum_{k=1}^{K-1}\sum_{l=k+1}^{K}\sum_{j=1}^{d}\big(r_{m,j}^{(k)} - r_{m,j}^{(l)}\big)^2. \tag{22}$$

Each method is assigned a stability weight using a softmax normalization:

$$\omega_m = \frac{\exp(\gamma \rho_m)}{\sum_{\ell=1}^{4} \exp(\gamma \rho_\ell)}, \quad 1 \le m \le 4, \tag{23}$$

where, $\gamma > 0$ controls the sensitivity to stability differences. Larger values emphasize the most stable method, whereas smaller values lead to nearly equal weights (i.e., $\exp(\gamma \rho_m)/4$).

Leveraging Eq. (23), the final stability-weighted consensus score is defined as Eq. (24):

$$\overline{\mathcal{S}}_{j,w} = \sum_{m=1}^{4} \omega_m \, \bar{S}_j^{(m,n)}, \tag{24}$$

where, $\bar{S}_j^{(m,n)}$ is the average over the K folds of the mth score, assessed using Eq. (8), Eq. (12), Eq. (14), and Eq. (18) as [see Eq. (25)]:

$$\overline{S}_j^{(m,n)} = \frac{1}{K} \sum_{k=1}^{K} S_{j,k}^{(m,n)}, \quad 1 \le m \le 4. \tag{25}$$

It is noteworthy that:

- Since $\omega_m \ge 0$, $\sum_{m=1}^{4} \omega_m = 1$, and $\bar{S}_j^{(m)} \in [0,1]$, it follows that $\overline{\mathcal{S}}_{j,w} \in [0,1]$;
- Methods exhibiting low ranking stability (small ρ_m) receive smaller weights ω_m through the softmax normalization, thereby reducing their influence on the final consensus score.

Subsequently, features are ranked according to $\overline{\mathcal{S}}_{j,w}$. Denote $\bar{S}_{\alpha_j,w}$ the jth greatest final stability-weighted score, i.e., Eq. (26):

$$\bar{S}_{\alpha_1,w} \ge \bar{S}_{\alpha_2,w} \ge \cdots \ge \bar{S}_{\alpha_d,w}, \quad \{\alpha_j\}_{j \le d} \in \{1,\ldots,d\}. \tag{26}$$

Lastly, the number of retained features d^* is determined using an elbow-based adaptive threshold [see Eq. (27)]:

$$d^* = \min\{j \in \{2,\ldots,d-1\} : |\Delta \bar{S}_{\alpha_j,w}| > \eta\}, \tag{27}$$

where,

$$\Delta \bar{S}_{\alpha_j,w} = \bar{S}_{\alpha_{j+1},w} - 2\,\bar{S}_{\alpha_j,w} + \bar{S}_{\alpha_{j-1},w}, \tag{28}$$

and $\eta > 0$ is a small tolerance parameter [see Eq. (28)].

Algorithm 1 summarizes the aforementioned steps of the proposed SW-MCFS process, which produces the top-d^* features from the selected subset $\mathcal{F}$. In practice, the parameter γ is fixed to 5 in all experiments.

Algorithm 1 SW-MCFS: Stability-Weighted Multi-Criteria Feature Selection

Input: $K, N, d, \mathcal{D}, \gamma, \eta, R$
Output: $\mathcal{F}^*$
$\mathcal{C} \leftarrow \lfloor \frac{N}{K} \rfloor$ //Cardinal of $\mathcal{D}_{\text{val}}^{(k)}$
for $k = 1$ to K **do**
//Split $\mathcal{D}$ into $\mathcal{D}_{\text{val}}^{(k)}$ and $\mathcal{D}_{\text{tr}}^{(k)}$ using (3)-(4)
  $\mathcal{D}_{\text{val}}^{(k)} \leftarrow \{(\mathbf{x}_{(k-1)\times\mathcal{C}+i}, y_i)\}_{i=1}^{\mathcal{C}}$
  $\mathcal{D}_{\text{tr}}^{(k)} \leftarrow \mathcal{D} \setminus \mathcal{D}_{\text{val}}^{(k)}$
  for $m = 1$ to 4 **do**
    for $j = 1$ to d **do**
      Evaluate $S_{j,k}^{(m,n)}$ // using Eqs. (8),(12),(14),(18)
    end for
    Compute $\overline{S}_j^{(m,n)}$ //using Eq. (25)
    Deduce $\mathbf{r}_m^{(k)}$ //defined in Eq. (20)
  end for
end for
//Stability Computation
for $m = 1$ to 4 **do**
  Evaluate ρ_m and ω_m //Using Eq. (19) and (23)
end for
for $j = 1$ to d **do**
  Evaluate $\bar{S}_{j,w}$ // Using Eq. (24)
end for
//Adaptive Threshold
Sort features in descending order of $\{\bar{S}_{j,w}\}_{1 \le j \le d}$
$j = 2$
repeat
  Compute $\Delta \bar{S}_{\alpha_j,w}$ //using Eq. (28)
  $j \leftarrow j + 1$
until $\Delta \bar{S}_{\alpha_{j-1},w} \le \eta$ **and** $j \le d$
$d^* \leftarrow j - 1$
$\mathcal{F}^* = \{\alpha_1, \ldots, \alpha_{d^*}\}$
return $\mathcal{F}^*$

D. Adaptive PSO (APSO)

After SW-MCFS reduces the original feature space to a lower-dimensional space of size $d^* \le d$, optimization is performed using a swarm of $\mathcal{S}$ particles. To balance exploration and exploitation adaptively, the concept of normalized swarm diversity is introduced within the APSO framework. The following notation formally defines the optimization procedure.

1) Notation: At iteration $t = 1, \ldots, \mathcal{T}_{\text{APSO}}$, particle $i = 1, \ldots, \mathcal{S}$ is characterized by:

- $\mathbf{p}_i^{(t)} \in \mathbb{R}^{d^*}$: *position vector* of particle i, representing a candidate set of model parameters.
  It is initialized as Eq. (29):
$$\mathbf{p}_i^{(0)} \sim \mathcal{U}(-p_{\max}, p_{\max})^{d^*}. \tag{29}$$
- $\mathbf{v}_i^{(t)} \in \mathbb{R}^{d^*}$: *velocity vector* controlling the update direction and exploration strength. It is initialized as Eq. (30):
$$\mathbf{v}_i^{(0)} \sim \mathcal{U}(-v_{\max}, v_{\max})^{d^*}. \tag{30}$$
- $\mathbf{p}_{i,\text{best}}^{(t)}$: personal best position of particle i, defined as the best position visited up to iteration t:
$$\mathbf{p}_{i,\text{best}}^{(t)} = \arg\min_{\tau \le t} f(\mathbf{p}_i^{(\tau)}), \qquad 1 \le i \le \mathcal{S}.$$
- $f(\mathbf{p})$: *fitness function* evaluating the validation performance of a particle [see Eq. (31)],
$$f(\mathbf{p}) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x,y) \in \mathcal{D}_{\text{val}}} \mathcal{L}(F(x;\mathbf{p}), y), \tag{31}$$

where, $F(x;\mathbf{p})$ denotes the NN parameterized by $\mathbf{p}$, and $\mathcal{L}$ is the loss function measuring the discrepancy between the predicted output $\hat{y} = F(x;\mathbf{p})$ and the true label y.

- $\mathbf{g}^{(t)}$: *global best position* among all particles [see Eq. (32)],

$$\mathbf{g}^{(t)} = \arg\min_{1\le i\le \mathcal{S}} f(\mathbf{p}_{i,\text{best}}^{(t)}). \tag{32}$$

The overall optimization problem is formulated as Eq. (33):

$$\mathbf{p}^* = \arg\min_{\mathbf{p}\in\mathbb{R}^{d^*}} f(\mathbf{p}). \tag{33}$$

2) Fitness-Landscape-Aware Inertia (FLAI): Instead of using a fixed inertia weight, it is adapted according to the current behavior of the swarm [21], [22].

The improvement of the global best at iteration t is defined as Eq. (34):

$$\Delta^{(t)} = \frac{f(\mathbf{g}^{(t-1)}) - f(\mathbf{g}^{(t)})}{f(\mathbf{g}^{(t-1)}) + \varepsilon}, \tag{34}$$

where, ε avoids numerical instability.

Swarm diversity is quantified by the dispersion of personal best positions [see Eq. (35)]:

$$\mathcal{F}^{(t)} = \frac{1}{\mathcal{S}\times d^*} \sum_{i=1}^{\mathcal{S}} \|\mathbf{p}_{i,\text{best}}^{(t)} - \bar{\mathbf{p}}_{\text{best}}^{(t)}\|_2, \tag{35}$$

where [see Eq. (36)],

$$\bar{\mathbf{p}}_{\text{best}}^{(t)} = \frac{1}{\mathcal{S}} \sum_{i=1}^{\mathcal{S}} \mathbf{p}_{i,\text{best}}^{(t)}. \tag{36}$$

Using the normalized versions $\hat{\mathcal{F}}^{(t)}$ and $\hat{\Delta}^{(t)}$ defined as Eq. (37):

$$\hat{\chi}^{(t)} = \frac{\chi^{(t)} - \chi_{\min}^{(t)}}{\chi_{\max}^{(t)} - \chi_{\min}^{(t)}}, \quad \chi \in \{\mathcal{F}, \Delta\}, \tag{37}$$

where Eq. (38),

$$\chi_{\min}^{(t)} = \min_{k\in[t-W+1,\,t]} \chi^{(k)}, \quad \chi_{\max}^{(t)} = \max_{k\in[t-W+1,\,t]} \chi^{(k)}, \tag{38}$$

and W is a sliding window size. As such, the inertia weight is updated as Eq. (39):

$$w^{(t)} = w_{\min} + (w_{\max} - w_{\min}) \times \mathcal{K}\big((\hat{\mathcal{F}}^{(t)} - \hat{\Delta}^{(t)}) \times \kappa\big), \tag{39}$$

with

$$\mathcal{K}(z) = \frac{1}{1+e^{-z}}.$$

Further, the parameter κ controls the sensitivity of the inertia update and initially $w^{(0)}$ is initialized as Eq. (40):

$$w^{(0)} = \frac{(w_{\max} + w_{\min})}{2}. \tag{40}$$

It is worth noting that:

- If $\hat{\mathcal{D}}_t \gg \hat{\Delta}_t$ (high diversity but weak improvement), the swarm explores without significant progress. In this case, the inertia weight decreases toward $w_{\min}$ to promote exploitation.
- If $\hat{\mathcal{D}}_t \ll \hat{\Delta}_t$ (low diversity but strong improvement), the swarm converges productively. The inertia weight is maintained at a moderate level to preserve stable convergence.
- If $\hat{\mathcal{D}}_t \approx \hat{\Delta}_t$, a balanced search state is achieved, leading to $w^{(t)} \approx (w_{\min} + w_{\max})/2$.
- If both $\hat{\mathcal{D}}_t$ and $\hat{\Delta}_t$ are small, stagnation is detected and stronger exploration mechanisms such as Lévy flight perturbation are triggered.

3) Lévy Flight Perturbation (LFP): When the swarm stagnates or loses diversity within τ_s consecutive iterations, randomness is injected using a Lévy flight to escape local minima. A β-index Lévy step is generated as [23] [see Eq. (41)]:

$$L_{j,\beta} = \frac{u_j}{|v_j|^{1/\beta}}, \quad j \le d^*, \tag{41}$$

where, u_j and v_j are two Gaussian distributions of zero mean [see Eq. (42)]:

$$u_j \sim \mathcal{N}(0, \sigma_u^2), \quad v_j \sim \mathcal{N}(0,1), \tag{42}$$

and

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\big(\frac{1+\beta}{2}\big)\,\beta\,2^{(\beta-1)/2}}\right]^{1/\beta},$$

where, $\Gamma(\cdot)$ is the Gamma function. Using the Gamma–Legendre duplication formula [52] $\Gamma(1+\beta)/\Gamma\big(\frac{1+\beta}{2}\big) = \frac{2^\beta}{\sqrt{\pi}}\Gamma\big(\frac{\beta}{2}+1\big)$ along with the identity $\Gamma\big(\frac{\beta}{2}+1\big) = \frac{\beta}{2}\Gamma\big(\frac{\beta}{2}\big)$, σ_u can be reduced to Eq. (43):

$$\sigma_u = \sqrt{2}\left[\frac{\sin(\pi\beta/2)}{\sqrt{2\pi}}\right]^{1/\beta}. \tag{43}$$

The particle is then perturbed around the current global best [see Eq. (44)]:

$$\mathbf{p}_i^{(t)} \leftarrow \mathbf{p}_i^{(t)} + \alpha_L\,\mathbf{L}_\beta \odot \big(\mathbf{p}_i^{(t)} - \mathbf{g}^{(t)}\big), \tag{44}$$

where, $\mathbf{L}_\beta = (L_{1,\beta}, \ldots, L_{d^*,\beta})^T$, $\odot$ denotes element-wise (Hadamard) multiplication, and α_L is a significant small step size scaling factor. This introduces long-range jumps when the search becomes trapped.

4) Elite-Guided Reinitialization (EGR): To further prevent premature convergence, every $\mathcal{T}_{\text{EGR}}$ iterations, the worst-performing $N_{\textbf{elite}}$ particles (i.e., with highest fitness) are reinitialized around the global elite using Gaussian-based EGR [see Eq. (45)]:

$$\mathbf{p}_{\vartheta_i}^{(t)} = \mathbf{g}_{\text{best}} + \boldsymbol{\xi}, \quad 1 \leq i \leq N_{\textbf{elite}}, \tag{45}$$

where [see Eq. (46)],

$$\boldsymbol{\xi} \sim \mathcal{N}\big(\mathbf{0},\, \sigma_{\text{EGR}}^2 \mathbf{I}_{d^*}\big)\,, \tag{46}$$

$$f(\mathbf{p}_{\vartheta_1}^{(t)}) \geq f(\mathbf{p}_{\vartheta_2}^{(t)}) \geq \ldots \geq f(\mathbf{p}_{\vartheta_{\mathcal{S}}}^{(t)}), \vartheta_i \in \{1 \ldots \mathcal{S}\}, \tag{47}$$

and $\sigma_{\text{EGR}} = 0.1 \times \mathcal{F}^{(t)}$. This step increases population diversity by injecting new candidate solutions near the best-known region while still preserving exploitation [see Eq. (47)].

5) APSO velocity and position update: Leveraging the inertia weight assessed in Eq. (39), the velocity is updated as Eq. (48):

$$\begin{aligned}\mathbf{v}_i^{(t)} = {} & w^{(t-1)}\mathbf{v}_i^{(t-1)} + c_1\mathbf{r}_1 \odot \big(\mathbf{p}_{i,\text{best}}^{(t-1)} - \mathbf{p}_i^{(t-1)}\big) \\ & + c_2\mathbf{r}_2 \odot \big(\mathbf{g}_{\text{best}}^{(t-1)} - \mathbf{p}_i^{(t-1)}\big),\end{aligned} \tag{48}$$

where,

- $\mathbf{g}_{\text{best}}^{(\tau)} = \min_{1 \leq \tau \leq t} \mathbf{g}^{(t)}$ is the global best position until t.
- c_1 and c_2 are the cognitive and social acceleration coefficients, respectively;
- $\mathbf{r}_1, \mathbf{r}_2 \sim \mathcal{U}(0,1)^{d^\star}$ are independent random vectors introducing stochastic exploration.

To prevent the uncontrolled growth of the velocity and to stabilize the search process, a component-wise bound is imposed as:

$$v_{i,d}^{(t)} = \min\Big(\max\big(v_{i,d}^{(t)}, -v_{\max}\big), v_{\max}\Big), \quad 1 \leq d \leq d^\star. \tag{49}$$

As such, the particle position is updated as follows [see Eq. (50)]:

$$\mathbf{p}_i^{(t)} = \mathbf{p}_i^{(t-1)} + \mathbf{v}_i^{(t)}, 1 \leq i \leq \mathcal{S}. \tag{50}$$

Similarly to Eq. (49), the positions are projected onto the feasible search space using a component-wise clipping operator [see Eq. (51)]:

$$p_{i,d}^{(t)} = \min\Big(\max\big(p_{i,d}^{(t)}, -p_{\max}\big), p_{\max}\Big), \quad 1 \leq d \leq d^\star. \tag{51}$$

Algorithm 2 presents the proposed APSO framework for global optimization. It iteratively updates particle velocities and positions using adaptive inertia, stochastic perturbation, and elite-guided reinitialization to enhance exploration and prevent premature convergence. The algorithm returns the optimal global best solution $\mathbf{g}_{\text{best}}$, which initializes the subsequent fine-tuning stage.

Algorithm 2 Adaptive PSO

Input: $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}, p_{\max}, v_{\max}, \mathcal{S}, \mathcal{T}_{\text{APSO}}, d^*, c_1, c_2, \kappa, \tau_s$ $\mathcal{T}_{\text{SGD}}, \varepsilon, W, w_{\min}, w_{\max}, \beta, \alpha_N, \sigma_{\text{EGR}}, \mathcal{T}_{\text{EGR}}, N_{\textbf{elite}}$
Output: $\mathbf{g}_{\text{best}}$ //Global best position
for $i = 1$ to $\mathcal{S}$ **do** //Initialize swarm position and speed
  Generate randomly $\mathbf{p}_i^{(0)}, \mathbf{v}_i^{(0)}$ //using Eqs. (29)-(30)
  Initialize $\mathbf{p}_{i,\text{best}} \leftarrow \mathbf{p}_i^{(0)}$
  Evaluate the fitness $f(\mathbf{p}_i^{(0)})$
  Initialize $Stag_i \leftarrow 0$ //Stagnation counters $\leq \tau_s$
end for
Deduce $\mathbf{g}^{(0)}$ //using Eq. (32)
Initialize $\mathbf{g}_{\text{best}} \leftarrow \mathbf{g}^{(0)}$
Compute σ_u //using Eq. (43)
Initialize inertia weight $\omega^{(0)}$ //using Eq. (40)
for $t = 1$ to $\mathcal{T}_{\text{APSO}}$ **do**
  for $i = 1$ to $\mathcal{S}$ **do**
    Generate two d^*-vectors $\mathbf{r}_1$ and $\mathbf{r}_2$
    Update $\mathbf{v}_i^{(t)}$ //using Eq. (48) and then (49)
    Update $\mathbf{p}_i^{(t)} \leftarrow \mathbf{p}_i^{(t-1)} + \mathbf{v}_i^{(t)}$ //As in (50) then (51)
    if $f(\mathbf{p}_i^{(t)}) < f(\mathbf{p}_{i,\text{best}})$ **then**
      $\mathbf{p}_{i,\text{best}} \leftarrow \mathbf{p}_i^{(t)}$; $Stag_i \leftarrow 0$
    else
      $Stag_i \leftarrow Stag_i + 1$
      if $Stag_i \geq \tau_s$ **then** //Apply LFP
        for $j = 1$ to d^* **do**
          Generate u_j, v_j //using (42)
          Deduce $L_{j,\beta}$ //using (41)
        end for
        Update $\mathbf{p}_i^{(t)}$ //using (44)
        Re-evaluate $f(\mathbf{p}_i^{(t)})$ and re-calculate $\mathbf{p}_{i,\text{best}}$
        $Stag_i \leftarrow 0$
      end if
    end if
    if $f(\mathbf{p}_{i,\text{best}}) < f(\mathbf{g}_{\text{best}})$ **then** //Update $\mathbf{g}_{\text{best}}$: (32)
      $\mathbf{g}_{\text{best}} \leftarrow \mathbf{p}_{i,\text{best}}$;
    end if
  end for
  Compute $\Delta^{(t)}$ //using Eq. (34)
  Evaluate $\bar{\mathbf{p}}_{\text{best}}^{(t)}$ and Deduce $\mathcal{F}^{(t)}$ //using (36),(35)
  Compute $\Delta_{\min}^{(t)}, \Delta_{\max}^{(t)}, \mathcal{F}_{\min}^{(t)}, \mathcal{F}_{\max}^{(t)}$ //using Eq. (37)
  Deduce $\hat{\Delta}^{(t)}, \hat{\mathcal{F}}^{(t)}$ //using Eq. (38)
  Deduce $w^{(t)}$ //using Eq. (39)
  //Apply EGR
  Set $\sigma_{\text{EGR}} \leftarrow 0.1 \times \mathcal{F}^{(t)}$
  if $t \bmod \mathcal{T}_{\text{EGR}} = 0$ **then**
    Sort $\mathbf{p}_i^{(t)}$ in descending order of fitness //as in (47)
    for $i = 1$ to N_{elite} **do**
      Generate randomly $\boldsymbol{\xi}$ //as in (46)
      Update $\mathbf{p}_{\vartheta_i}^{(t)}$ //as in (45)
      if $f(\mathbf{p}_{\vartheta_i}^{(t)}) < f(\mathbf{p}_{\vartheta_i,\text{best}})$ **then**
        $\mathbf{p}_{\vartheta_i,\text{best}} \leftarrow \mathbf{p}_{\vartheta_i}^{(t)}$
      end if
    end for
  end if
  Recompute $\mathbf{g}_{best}$ from $\mathbf{p}_{\vartheta_i,\text{best}}$
  //Save position and velocity for subsequent iteration
  $\mathbf{p}_i^{(t+1)} \leftarrow \mathbf{p}_i^{(t)}, \quad \mathbf{v}_i^{(t+1)} \leftarrow \mathbf{v}_i^{(t)}$
end for
return $\mathbf{g}_{\text{best}}$

6) SGD-Based fine-tuning: After the APSO stage described in Section III-D, the optimal solution $\mathbf{g}_{\text{best}} \in \mathbb{R}^{d^*}$ is reshaped to initialize the parameters of a feedforward neural network with architecture $[d^*, n_1, 1]$, where d^* is the input dimension and n_1 denotes the number of hidden neurons. As such, each training sample is represented by a reduced feature vector $\mathbf{s} \in \mathbb{R}^{d^*}$ and $\mathbf{g}_{\text{best}}$ is partitioned into structured parameter blocks [see Eq. (52)]:

$$\mathbf{g}_{\text{best}} = [\mathbf{g}_{W_1}; \mathbf{g}_{b_1}; \mathbf{g}_{w_2}; g_{b_2}], \tag{52}$$

where, $\mathbf{g}_{W_1} \in \mathbb{R}^{n_1 \times d^*}$, $\mathbf{g}_{b_1} \in \mathbb{R}^{n_1}$, $\mathbf{g}_{w_2} \in \mathbb{R}^{n_1}$, and $g_{b_2} \in \mathbb{R}$.

The forward propagation step computes the hidden representation via a linear transformation followed by a ReLU activation. The output layer applies the sigmoid activation, defined in Section III-D2, to produce the final prediction [see Eq. (53)]:

$$\hat{y} = \mathcal{K}(\mathbf{g}_{w_2}^\top \mathbf{h} + g_{b_2}), \tag{53}$$

where [see Eq. (54)],

$$\mathbf{h} = \max(\mathbf{0}, \mathbf{g}_{W_1}\mathbf{s} + \mathbf{g}_{b_1}), \tag{54}$$

The complete trainable parameter vector is defined as Eq. (55):

$$\boldsymbol{\theta} = \{\mathbf{g}_{W_1}; \mathbf{g}_{b_1}; \mathbf{g}_{w_2}; g_{b_2}\}, \tag{55}$$

initialized at Eq. (56):

$$\boldsymbol{\theta}_0 = \text{reshape}(\mathbf{g}_{\text{best}}). \tag{56}$$

Obviously, $\mathbb{R}^{|\boldsymbol{\theta}|} = n_1 d^* + 2n_1 + 1$. The model is trained via backpropagation [46] by minimizing the binary cross-entropy loss with L_2 regularization:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{j=1}^{|\mathcal{D}_{\text{tr}}|} \underbrace{[-y_j \log \hat{y}_j - (1-y_j)\log(1-\hat{y}_j)]}_{\triangleq \mathcal{E}_j} - \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \tag{57}$$

where, $|\mathcal{D}_{\text{tr}}|$ is the number of training samples and λ controls the regularization strength.

At each iteration, the gradient of the loss function in Eq. (57) is computed over a mini-batch $\mathcal{B} \subset \mathcal{D}_{\text{tr}}$ of fixed size $|\mathcal{B}|$ [see Eq. (58)]:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \nabla_{\boldsymbol{\theta}} \mathcal{E}_j + \lambda \boldsymbol{\theta}, \tag{58}$$

where, ∇ denotes the gradient operator and $\mathcal{E}_j$ denotes the binary cross-entropy loss for sample j.

Optimization is performed using stochastic gradient descent [34], [35] with momentum [47]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathbf{u}_t, 0 \le t < \mathcal{T}_{\text{SGD}} - 1, \tag{59}$$

where, $\mathbf{u}_t \in \mathbb{R}^{|\boldsymbol{\theta}|}$, defined as:

$$\begin{cases} \mathbf{u}_t = \mu_m \mathbf{u}_{t-1} + \eta_\ell \nabla_{\boldsymbol{\theta}} \mathcal{L}, & 1 \le t \le \mathcal{T}_{\text{SGD}} \\ \mathbf{u}_0 = \mathbf{0} \end{cases}. \tag{60}$$

where, η_ℓ is the learning rate and μ_m is the momentum coefficient. Mini-batch training with a fixed batch size is used, and early stopping based on validation performance prevents overfitting.

Thus, the SGD stage iteratively applies the parameter update in Eq. (59) with momentum accumulation as defined in Eq. (60) to refine the APSO-initialized parameters and obtain the final optimized model.

Algorithm 3 summarizes the complete fine-tuning procedure.

Algorithm 3 SGD Fine-Tuning

Input: $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}, \mathbf{g}_{\text{best}}, \lambda, \eta_\ell, \mu_m, T_{\text{SGD}}, |\mathcal{B}|, T_{\text{pat}}$
Output: $\boldsymbol{\theta}^*$
$\boldsymbol{\theta}_0 \leftarrow \text{reshape}(\mathbf{g}_{\text{best}})$ //as in (55)
$\mathbf{u}_0 \leftarrow \mathbf{0}$
$\mathcal{L}^* \leftarrow \infty$, wait $\leftarrow 0$
for epoch $= 1$ to $\mathcal{T}_{\text{SGD}}$ **do**
  for each mini-batch $\mathcal{B}$ in epoch **do** // $\left\lfloor \frac{|\mathcal{T}_{\text{SGD}}|}{|\mathcal{B}|} \right\rfloor$ times
    $\mathbf{g}_t \leftarrow \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \nabla_{\boldsymbol{\theta}} \mathcal{L} + \lambda \boldsymbol{\theta}$
    $\mathbf{u}_t \leftarrow \mu_m \mathbf{u}_{t-1} + \eta_\ell \mathbf{g}_t$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathbf{v}_t$
  end for
  $\mathcal{L}_{\text{val}} \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_{\text{val}})$
  if $\mathcal{L}_{\text{val}} < \mathcal{L}^*$ **then**
    $\mathcal{L}^* \leftarrow \mathcal{L}_{\text{val}}$
    $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
    wait $\leftarrow 0$
  else
    wait $\leftarrow$ wait $+ 1$
    if wait $\ge T_{\text{pat}}$ **then**
      break
    end if
  end if
end for
return $\boldsymbol{\theta}^*$

IV. Experimental Setup and Data Description

This section describes the dataset, preprocessing pipeline, and experimental configuration used to evaluate the proposed framework. All preprocessing steps were performed using the training data only to avoid information leakage.

A. Dataset Description

The dataset contains $10,247$ recruitment records collected from a telecommunications company in North Africa between 2018 and 2022. After removing duplicates and incomplete evaluations (minimum six months of employment), the final dataset was retained for analysis.

The data includes 14 features, as outlined in Table I, grouped into four categories:

- Demographic (e.g., age, gender, marital status),
- Educational (e.g., education level, specialization, certifications),
- Professional (e.g., experience, salary, seniority, language proficiency),

TABLE I. DATASET DESCRIPTION GROUPED BY FEATURE CATEGORIES ($n = 10{,}247$)

Category	Feature	Value Range / Distribution
Demographic Features		
	Age	22–58 (mean: 31.4, SD: 6.2)
	Gender	Male (54%) / Female (46%)
	Marital Status	Single (61%) / Married (39%)
Educational Features		
	Education Level	Bachelor (58%), Master (35%), PhD (7%)
	Specialization	Tech (42%), Econ (31%), HR (15%), Law (12%)
	Certifications	Yes (34%) / No (66%)
Professional Features		
	Experience	0–15 years (mean: 4.8, SD: 3.1)
	Salary	4,000–18,000 MAD (mean: 8,200, SD: 2,800)
	Language Proficiency	1–5 (mean: 3.2, SD: 1.1)
Behavioral Features		
	Communication	1–5 (mean: 3.6, SD: 0.9)
	Motivation	Yes (73%) / No (27%)
	Remote Work	Yes (68%) / No (32%)
	Physical Ability	Yes (91%) / No (9%)
Target Variable		
	Performance	BA (38%) / Good (62%)

- Behavioral (e.g., communication skills, motivation, remote work adaptability, physical ability).

The target variable represents job performance and is defined as a binary label: *Below Average (BA)* and *Good*.

It should be noted that binary performance labels derived from HR evaluations are inherently subject to noise, as they reflect subjective managerial assessments that may vary across evaluators and organizational contexts. This label noise imposes an upper bound on achievable predictive accuracy and should be considered when interpreting the reported results. The 76.8% accuracy achieved by the proposed framework may partly reflect this noise ceiling rather than a fundamental model limitation.

Ethical approval was obtained from the institutional review board. All personal identifiers were removed, and data handling followed privacy regulations.

B. Data Preprocessing

The dataset was divided into 80% training ($n = 8{,}197$) and 20% test ($n = 2{,}050$) using stratified sampling. A 10-fold stratified cross-validation procedure was applied on the training set.

The preprocessing pipeline consists of the following steps:

- *Missing values:* Missing entries in several attributes are estimated using KNN imputation [37], [38] with k_{nn} neighbors. For a missing value $w_{i,j}$, the estimate is computed as Eq. (61):

$$\hat{x}_{i,j} = \frac{1}{k_{\text{nn}}} \sum_{\ell \in \mathcal{N}_{k_{\text{nn}}}(i)} x_{\ell,j}, \tag{61}$$

  where, $\mathcal{N}_{k_{\text{nn}}}(i)$ denotes the set of the k_{nn} nearest samples to observation i.

- *Encoding:* Binary variables are encoded as $\{0, 1\}$, while categorical attributes with multiple categories are transformed using one-hot encoding.

- *Scaling:* Continuous variables are normalized using Min–Max scaling [see Eq. (62)]:

$$x_{i,j}^{\text{scaled}} = \frac{x_{i,j} - x_j^{(\min)}}{x_j^{(\max)} - x_j^{(\min)}}. \tag{62}$$

- *Outlier handling:* Extreme values are capped using the interquartile range (IQR) rule [40], affecting approximately 1.4% of the observations.

All transformations were fitted on training data and applied to validation and test sets to prevent data leakage.

C. Baseline Models and Configurations

Classical machine learning models are used as benchmarks to evaluate the effectiveness of the proposed framework, including RF [18], SVM [49], and gradient boosting methods [26], [41]. The standard ANN baseline employs the Adam optimizer [36] with dropout regularization [45]. Xavier initialization [48] is used for SGD-only baselines. All implementations use scikit-learn [39] and custom Python modules. Rather than detailing their internal formulations, their feature usage, architectures, and optimization strategies are summarized in Table II.

TABLE II. MODEL CONFIGURATIONS USED FOR BENCHMARKING

Model	*Features*	*Architecture*	*Optimization*
RF	14	500 trees	–
XGBoost	14	500 trees	–
SVM	14	RBF kernel	SMO
Std. ANN	14	$(14 - 21 - 1)$	Adam
LR-ANN+SGD	LR (12)	$(12 - 18 - 1)$	SGD
LR-ANN+PSO-SGD	LR (12)	$(12 - 18 - 1)$	PSO $\rightarrow$ SGD
SW-MCFS-APSO-SGD	**SW-MCFS**	**Adaptive**	**APSO $\rightarrow$ SGD**

D. Experimental Setup

This subsection summarizes the experimental configuration used to evaluate the proposed framework. All hyperparameters for Algorithm 1 to Algorithm 3 are reported in Table III. These settings were fixed across all experiments to ensure fair comparison and reproducibility.

The evaluation protocol is based on:

- *Cross-validation:* 10-fold stratified cross-validation on the training set.
- *Performance metrics:* Accuracy, Precision, Recall, Specificity, and F1-Score.
- *Statistical testing:* McNemar's test for pairwise comparison on the held-out test set and the Friedman test with Nemenyi post-hoc analysis across folds. The significance level is set to $\alpha = 0.05$.
- *Ablation analysis:* Each ablation configuration is repeated 10 times with different random seeds to assess stability and robustness.

V. RESULTS

Comprehensive experiments are conducted to assess feature selection quality, predictive performance, statistical significance, robustness, interpretability, and computational efficiency of the proposed framework.

TABLE III. PARAMETER SETTINGS FOR ALGORITHM 1, 2, AND 3

Alg.	*Param.*	*Val.*	*Param.*	*Val.*	*Param.*	*Val.*
1	K	10	N	10247	d	14
	γ	5	η	0.05	R	30
2	$p_{\max}$	2	$v_{\max}$	1	$\mathcal{T}_A$	200
	$\mathcal{S}$	20	d^*	10	ϵ	10^{-10}
	W	5	κ	10	$w_{\min}$	0.2
	$w_{\max}$	0.9	β	1.5	τ_s	5
	α_L	0.01	N_e	$\lfloor \mathcal{S}/5 \rfloor$	$\mathcal{T}_E$	10
	c_1	2.0	c_2	2.0	η_ℓ	0.01
	μ_m	0.9	$\lvert\mathcal{B}\rvert$	32	T_{pat}	15

A. Feature Selection Analysis

1) Method stability and consensus weights: Table IV reports the cross-validation stability scores ρ_m and the corresponding consensus weights ω_m computed for each feature scoring method.

TABLE IV. STABILITY AND CONSENSUS WEIGHTS FOR SW-MCFS

Method	ρ_m	ω_m
Mutual Information	0.91	0.31
Wald Significance	0.87	0.24
Fisher Discriminant Loading	0.84	0.20
Permutation Importance	0.89	0.25

Mutual Information achieves the highest stability, resulting in the largest weight. In contrast, Fisher Discriminant Loading exhibits the lowest stability and is automatically down-weighted. This demonstrates that the proposed aggregation mechanism effectively prioritizes reliable scoring methods.

2) Feature relevance and selection outcome: Table V presents the normalized feature relevance scores and the aggregated score $\overline{S}_{\alpha_j,w}$.

TABLE V. REDUCED $d^* = 10$ FEATURES USING THE SW-MCFS ALGORITHM.

j	*Feature*	$\overline{S}^{(m,n)}_{1,\alpha_j}$	$\overline{S}^{(m,n)}_{2,\alpha_j}$	$\overline{S}^{(m,n)}_{3,\alpha_j}$	$\overline{S}^{(m,n)}_{4,\alpha_j}$	$\overline{S}_{\alpha_j,w}$
1	*Communication*	1.00	0.92	0.88	1.00	**0.96**
2	*Experience*	0.87	1.00	0.91	0.82	**0.90**
3	*Seniority*	0.82	0.85	1.00	0.78	**0.86**
4	*Salary*	0.79	0.78	0.82	0.85	**0.81**
5	*Education*	0.71	0.74	0.69	0.73	**0.72**
6	*Language Prof.*	0.68	0.65	0.61	0.71	**0.67**
7	*Specialization*	0.62	0.58	0.55	0.64	**0.60**
8	*Remote Work*	0.55	0.52	0.48	0.58	**0.54**
9	*Certifications*	0.48	0.51	0.44	0.46	**0.47**
10	*Age*	0.42	0.45	0.38	0.40	**0.41**
11	*Gender*	0.28	0.31	0.22	0.25	0.27
12	*Marital Status*	0.24	0.27	0.19	0.21	0.23
13	*Motivation*	0.18	0.15	0.12	0.20	0.16
14	*Physical Ability*	0.12	0.08	0.14	0.11	0.11

Using the adaptive elbow threshold defined in Eq. (27) with $\tau^* = 0.34$, the algorithm selects $d^* = 10$ features. Sensitive demographic attributes such as Gender and Marital Status are automatically discarded due to their low aggregated relevance.

Compared with baseline selection methods, SW-MCFS preserves informative predictors (e.g., Experience, Education, Communication) while filtering low-signal attributes. This improves both interpretability and robustness against potential bias.

TABLE VI. PERFORMANCE COMPARISON (10-FOLD CV, $n = 8{,}197$) MEAN $\pm$ SD

Model	*Acc.*	*Prec.*	*Rec.*	*Spec.*	*F1*
Random Forest	72.1 ± 1.8	70.3 ± 2.1	73.5 ± 2.4	69.8 ± 2.0	71.8 ± 1.9
XGBoost	73.8 ± 1.5	72.1 ± 1.9	75.2 ± 2.0	71.4 ± 1.7	73.6 ± 1.6
SVM	70.4 ± 2.3	68.9 ± 2.5	71.2 ± 2.8	69.1 ± 2.4	70.0 ± 2.3
Standard ANN	73.2 ± 1.9	71.5 ± 2.2	74.6 ± 2.1	70.9 ± 2.0	73.0 ± 1.9
LR-ANN + SGD	74.6 ± 1.6	73.1 ± 1.8	75.6 ± 1.9	72.8 ± 1.7	74.3 ± 1.6
LR-ANN + PSO-SGD	75.2 ± 1.4	73.6 ± 1.7	76.3 ± 1.8	73.5 ± 1.6	74.9 ± 1.5
SW-MCFS-ANN + SGD	76.2 ± 1.3	74.8 ± 1.6	77.1 ± 1.7	74.5 ± 1.5	75.9 ± 1.4
SW-MCFS-APSO-SGD	$\mathbf{76.8 \pm 1.2}$	$\mathbf{75.3 \pm 1.5}$	$\mathbf{77.8 \pm 1.6}$	$\mathbf{75.2 \pm 1.4}$	$\mathbf{76.5 \pm 1.3}$

B. Model Comparison Results

Table VI compares the predictive performance of classical machine learning models and neural networks under different optimization strategies. Performance is evaluated using Accuracy, Precision, Recall, Specificity, and F1-score. Results are reported as mean $\pm$ standard deviation (SD) over 10-fold stratified cross-validation.

The proposed SW-MCFS-APSO-SGD framework achieves the highest accuracy while maintaining the lowest variance across folds. The reduced standard deviation indicates stable convergence and strong generalization ability.

C. Statistical Significance Analysis

To verify whether performance improvements are statistically significant, pairwise and global non-parametric tests are performed. Table VII reports the McNemar's test results comparing the proposed SW-MCFS-APSO-SGD against all competing models.

TABLE VII. MCNEMAR'S TEST COMPARING SW-MCFS-APSO-SGD AGAINST COMPETING MODELS ($p < 0.05$).

Comparison (vs.)	χ^2	*p-value*
Random Forest	16.42	< 0.001
XGBoost	6.31	0.012
SVM	22.87	< 0.001
Standard ANN	8.14	0.004
LR-ANN + SGD	4.85	0.028
LR-ANN + PSO-SGD	4.18	0.041
SW-MCFS-ANN + SGD	3.92	0.048

The Friedman test yields $\chi^2(7) = 45.23$ with $p < 0.001$, confirming overall statistical differences among models. Post-hoc Nemenyi analysis further ranks the proposed framework as the best-performing method with a critical difference of 1.42 at $\alpha = 0.05$.

These results confirm that the improvements of the proposed approach are statistically significant.

D. Ablation Studies

Ablation experiments are conducted to quantify the contribution of individual components.

1) Ablation study for feature selection methods: Table VIII reports the performance of different feature selection strategies, highlighting the contribution of the proposed SW-MCFS method compared to individual and baseline selection techniques.

TABLE VIII. ABLATION A1: FEATURE SELECTION COMPARISON (ALL WITH SGD OPTIMIZER).

Selection Method	*# feat.*	*Accuracy*	*F1*
MI only	11	74.8 ± 1.5	74.5
Wald (LR) only	12	74.6 ± 1.6	74.3
FDL (DA) only	8	68.5 ± 2.1	67.7
PI only	11	75.0 ± 1.5	74.7
Equal-weight consensus	10	75.8 ± 1.4	75.5
SW-MCFS (ours)	**10**	$\mathbf{76.2 \pm 1.3}$	**75.9**

SW-MCFS outperforms single-method selection and equal-weight aggregation. The stability-based weighting contributes to consistent improvements in predictive accuracy.

2) Optimization ablation: Table IX presents the contribution of each component in the proposed framework, evaluating the performance degradation when individual modules are removed.

TABLE IX. ABLATION A2: OPTIMIZER COMPARISON (ALL WITH SW-MCFS FEATURES).

Optimizer	*Accuracy*	*Epochs to 95%*	*Time (s)*
SGD only (Xavier)	76.2 ± 0.8	50.2 ± 4.3	138 ± 8
PSO $\rightarrow$ SGD	76.6 ± 0.6	38.1 ± 3.7	187 ± 12
APSO $\rightarrow$ SGD	$\mathbf{76.8 \pm 0.5}$	$\mathbf{34.5 \pm 3.1}$	$\mathbf{195 \pm 14}$

APSO improves convergence speed and final accuracy compared with standard PSO and SGD-only training with Xavier initialization [48]. The hybrid strategy achieves better optimization stability.

3) Component contribution: Table X quantifies the effect of removing individual components from the proposed framework to assess their relative contribution to the overall performance.

TABLE X. CONTRIBUTION OF INDIVIDUAL COMPONENTS

Configuration	*Accuracy*	Δ *vs. full*
LR + SGD (baseline)	74.6	-2.2
LR + PSO-SGD	75.2	-1.6
SW-MCFS + SGD	76.2	-0.6
SW-MCFS + PSO-SGD	76.6	-0.2
SW-MCFS + APSO-SGD (full)	**76.8**	—

Both feature selection and optimization contribute positively and additively to the overall performance improvement.

E. SHAP Feature Importance

Table XI reports the global SHAP importance scores [50] computed on the trained SW-MCFS-APSO-SGD model, ranking features according to their mean absolute contribution.

TABLE XI. GLOBAL SHAP IMPORTANCE FOR THE PROPOSED MODEL. DEMOGRAPHIC ATTRIBUTES SUCH AS GENDER AND MARITAL STATUS ARE EXCLUDED BY SW-MCFS.

Feature	*Mean* $\lvert\phi_i\rvert$	*Rank*
Communication Skills	0.358	1
Experience	0.301	2
Seniority Level	0.284	3
Salary	0.259	4
Education Level	0.215	5
Language Proficiency	0.194	6
Specialization	0.172	7
Remote Work	0.148	8
Certifications	0.078	9
Age	0.065	10

The results confirm that SW-MCFS automatically excludes sensitive demographic attributes, while communication skills, experience, and seniority remain dominant predictors.

F. Held-Out Test Set Performance

Table XII compares the proposed framework against baseline models on the independent held-out test set.

TABLE XII. PERFORMANCE ON THE INDEPENDENT TEST SET ($n = 2{,}050$)

Model	*Acc.*	*Prec.*	*Rec.*	*F1*
Random Forest	71.8	69.9	73.2	71.5
XGBoost	73.5	71.8	74.9	73.3
SVM	70.1	68.5	70.8	69.6
Standard ANN	72.9	71.2	74.3	72.7
LR-ANN + PSO-SGD	75.0	73.4	76.0	74.7
SW-MCFS-APSO-SGD	**76.5**	**75.1**	**77.6**	**76.3**

The proposed method achieves the highest performance across all metrics, demonstrating strong generalization capability.

G. Computational Efficiency

Table XIII reports the training and inference time of the proposed framework compared to competing models.

TABLE XIII. TRAINING INCLUDES SW-MCFS COMPUTATION ($\sim$24S). IDENTICAL INFERENCE FOR ALL ANNS.

Model	*Train (s)*	*Inference (s/1k)*
XGBoost	68	0.08
LR-ANN + PSO-SGD	187	0.05
SW-MCFS + SGD	162	0.05
SW-MCFS-APSO-SGD	**219**	**0.05**

Although the proposed model requires slightly higher training time due to APSO optimization, inference complexity remains unchanged while achieving superior performance.

VI. DISCUSSION

A. Principal Findings

This study highlights four key findings:

1) SW-MCFS significantly improves feature selection: The stability-weighted multi-criteria approach improves accuracy by 1.6% over single-method LR selection (Table X). Its effectiveness stems from aggregating heterogeneous relevance criteria, applying stability-based weighting to suppress unreliable scorers, and automatically determining the feature dimension via an adaptive threshold. It is important to emphasize that the novelty lies not in the individual scoring methods, which are well-established, but in their principled integration through stability-weighted consensus—a combination that produces emergent benefits including automatic dimensionality determination and demographic attribute filtering that no single method achieves independently.

2) APSO enhances optimization efficiency: The adaptive inertia strategy improves the exploration–exploitation balance, yielding 0.6% gain over SGD-only and 0.2% over standard PSO, with 31% faster convergence. Lévy-flight perturbations further improve robustness by enabling escape from local optima.

3) Observed exclusion of demographic attributes: SW-MCFS systematically excludes Gender and Marital Status, producing a feature set focused on job-relevant variables. This behavior emerges from consensus scoring rather than explicit constraints, as unstable and low-importance features are downweighted across folds. However, this observation should not be interpreted as a formal fairness guarantee; formal fairness metrics such as demographic parity and equalized odds were not computed in this study and remain a necessary direction for future validation [31].

4) Competitive performance against ensemble models: The proposed SW-MCFS–APSO–SGD framework achieves 76.8% accuracy, significantly outperforming XGBoost (73.8%, $p = 0.012$), demonstrating that well-designed neural pipelines can surpass strong ensemble baselines on structured HR data. It is worth noting that in predictive hiring, where target labels are derived from subjective performance evaluations, the inherent label noise imposes an accuracy ceiling that limits all models. Within this context, a 3% absolute improvement over XGBoost represents a meaningful and statistically significant gain [43].

B. Novelty Positioning

Table XIV contrasts the proposed framework with representative prior works. Existing studies typically address either multi-criteria feature selection or adaptive optimization separately. In contrast, the proposed framework jointly integrates stability-weighted multi-method feature selection with adaptive particle-based optimization in a unified learning pipeline.

TABLE XIV. POSITIONING OF THE PROPOSED FRAMEWORK RELATIVE TO PRIOR WORK.

Study	*Multi-FS*	*Adaptive PSO*	*HR Data*
Saeys et al. [20]	✓	–	–
Zhan et al. [22]	–	✓	–
Qin et al. [24]	–	–	✓
Wang et al. [25]	–	–	✓
Ours	✓	✓	✓

C. Limitations

Despite strong empirical performance, several limitations should be acknowledged.

1) Dataset scope: All experiments were conducted on a single-organizational dataset from a North African telecommunications company, which limits external generalizability. The abstract and claims throughout the study are qualified accordingly. Validation across multiple industries, geographic regions, and organizational cultures is required before broader applicability can be established.

2) Label quality: The binary performance labels are derived from subjective managerial evaluations, which constitute a significant source of noise in predictive hiring datasets. Inter-rater reliability and evaluation criteria consistency were not assessed. This noise ceiling contextualizes the 76.8% accuracy and suggests that absolute performance gains should be interpreted relative to this inherent limitation.

3) Fairness evaluation: Although demographic features (Gender, Marital Status) were automatically excluded by SW-MCFS due to low consensus scores, this observation does not constitute a formal fairness guarantee. No formal fairness metrics such as demographic parity, equalized odds, or disparate impact ratios were computed [31]. The exclusion of protected attributes from the feature set does not preclude indirect discrimination through proxy variables (e.g., salary or seniority may correlate with demographic characteristics). Formal fairness auditing remains a critical direction for deployment readiness.

4) Computational cost: SW-MCFS requires multiple feature scorers over K folds, and APSO introduces additional optimization overhead. Scalability to very large datasets (e.g., $> 100{,}000$ records) remains to be evaluated.

5) Hyperparameter sensitivity: Parameters such as γ, APSO settings, and population size were tuned empirically. A systematic sensitivity analysis would further strengthen robustness claims.

D. Ethical Considerations

The deployment of predictive hiring systems raises significant ethical concerns that extend beyond predictive accuracy. Algorithmic hiring tools are increasingly subject to regulatory scrutiny, including the European Union Artificial Intelligence Act [53], which classifies employment-related AI systems as high-risk and mandates transparency, human oversight, and bias auditing requirements [43].

Several ethical dimensions are relevant to the proposed framework. First, although SW-MCFS excludes Gender and Marital Status from the selected feature set, indirect discrimination may persist through correlated proxy variables. For instance, salary levels and seniority may encode historical gender-based disparities, enabling the model to implicitly learn demographic associations. Formal fairness auditing using established metrics (demographic parity, equalized odds, and disparate impact analysis) is essential before any operational deployment.

Second, the use of subjective performance labels as ground truth raises concerns about perpetuating existing organizational

biases. If historical evaluations systematically disadvantaged certain groups, the trained model may reproduce these patterns. Bias-aware label correction and calibration techniques should be explored in future work.

Third, transparency and explainability are critical for stakeholder trust. The SHAP-based interpretability analysis presented in this study provides a foundation for explaining individual predictions to candidates and hiring managers. However, the right to explanation and the principle of human-in-the-loop decision-making require that predictive scores serve as decision-support tools rather than automated gatekeepers.

E. Future Directions

Several extensions can enhance the proposed framework:

1) Fairness-aware extension: Incorporating an explicit fairness-aware scoring term into SW-MCFS would allow penalizing features strongly correlated with protected attributes [see Eq. (63)].

$$\overline{S}^{\text{fair}}_{\alpha_j,w} = \sum_m \omega_m \bar{S}^{(m)}_{\alpha_j} - \lambda_f \mathcal{F}_j, \tag{63}$$

where, λ_f controls the fairness–accuracy trade-off.

2) Deep architectures: Applying the SW-MCFS–APSO pipeline to deeper tabular models (e.g., TabNet [51] or transformer-based architectures) would test scalability to higher-dimensional parameter spaces.

3) Cross-domain validation: Benchmarking on public HR and structured prediction datasets (credit risk, medical classification, etc.) would further validate generalization capability.

4) Online feature selection: Developing an incremental version of SW-MCFS would enable dynamic feature relevance updates as new data becomes available.

VII. Conclusion

This study presents two complementary methodological contributions for predictive hiring: *Stability-Weighted Multi-Criteria Feature Selection (SW-MCFS)* and *Adaptive Particle Swarm Optimization (APSO)*. SW-MCFS aggregates heterogeneous feature scoring methods through stability-weighted consensus across cross-validation folds, producing robust feature subsets while automatically filtering low-signal and demographic attributes. APSO enhances neural network initialization by combining adaptive inertia control with Lévy-flight perturbations, enabling effective exploration of complex optimization landscapes.

The integrated SW-MCFS-APSO-SGD framework achieves 76.8% accuracy on 10,247 recruitment records, significantly outperforming XGBoost (73.8%, $p = 0.012$), standard PSO-SGD (75.2%, $p = 0.041$), and individual feature selection strategies. Ablation analyses confirm that SW-MCFS provides the largest gain ($+1.6\%$), while APSO further improves optimization performance ($+0.6\%$) and accelerates convergence by approximately 31%. These results should be interpreted in the context of inherent label noise in subjective performance evaluations, which imposes an accuracy ceiling on all models.

Notably, SW-MCFS excludes Gender and Marital Status through consensus scoring without explicit fairness constraints. While this observation is encouraging, it does not constitute a formal fairness guarantee, and formal fairness auditing remains necessary for deployment readiness. The SHAP-based interpretability analysis, which highlights Communication Skills, Experience, and Seniority as dominant predictors, provides a foundation for transparent decision-support in regulated HR applications.

The proposed framework is applicable to other classification tasks requiring robust feature selection and efficient neural optimization. Future work will extend the approach to multi-domain validation across diverse industries and geographic contexts, incorporate explicit fairness objectives with formal bias metrics, and explore scalability to deeper architectures and multimodal data.

Data Availability Statement

The dataset is derived from anonymized recruitment data collected at ENSIAS, Université Mohammed V – Rabat. Due to confidentiality agreements, the data cannot be publicly shared. Researchers may contact the corresponding author. Analysis scripts and SW-MCFS implementation are available upon reasonable request.

Author Contributions

Y.T.K. conceptualized the study, designed SW-MCFS and APSO, performed analysis, and drafted the manuscript. S.A. supervised, validated results, and revised the manuscript. Both authors approved the final version.

Funding

This research received no external funding.

Institutional Review Board Statement

Approved by the participating organization's internal review board (Protocol: HR-2023-015, 15 January 2023).

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgment

The authors thank the HR department of the participating organization and the anonymous reviewers for constructive feedback.

References

[1] Z.-H. Zhou, K. Chen, and X. Y. Dai, "Data-driven intelligence in hiring and employee selection," *Int. J. Data Sci. Anal.*, vol. 11, no. 3, pp. 221–234, 2021.

[2] R. Chamorro-Premuzic, D. Winsborough, R. A. Sherman, and R. Hogan, "New talent signals: Shiny new objects or a brave new world?" *Ind. Organ. Psychol.*, vol. 9, no. 3, pp. 621–640, 2016.

[3] S. Pessach, G. Singer, D. Avrahami, H. C. Ben-Gal, E. Shmueli, and I. Ben-Gal, "Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming," *Decis. Support Syst.*, vol. 134, p. 113290, 2020.

[4] V. Jagwani, S. Meghani, K. Pai, and S. Dhage, "Resume evaluation through LDA and NLP for effective candidate selection," *arXiv:2307.15752*, 2023.

[5] E. Faliagka, A. Tsakalidis, and G. Tzimas, "An integrated e-recruitment system for automated personality mining and applicant ranking," *Internet Res.*, vol. 22, no. 5, pp. 551–568, 2012.

[6] P. Levy, O. Golan, and A. Schuster, "Enhancing interview evaluations with nonverbal behavior analysis," *IEEE Trans. Affect. Comput.*, vol. 9, no. 2, pp. 205–216, 2018.

[7] S. Strohmeier and F. Piazza, "Artificial intelligence techniques in human resource management—A conceptual exploration," in *Intelligent Techniques in Engineering Management*. Cham: Springer, 2015, pp. 149–172.

[8] J. Pfeffer and R. I. Sutton, "Evidence-based management," *Harvard Bus. Rev.*, vol. 84, no. 1, pp. 62–74, 2006.

[9] IBM Smarter Workforce Institute, *The Business Case for AI in HR*. Armonk, NY: IBM Corp., 2017.

[10] Y. Khallouk Temsamani and S. Achchab, "Artificial intelligence use in human resources management: Strategy and operation's impact," ENSIAS, Mohammed V University, Rabat, Morocco, 2021.

[11] N. Otani *et al.*, "Natural language processing for human resources," in *Proc. NAACL Industry Track*, 2025.

[12] K. Khelkhal and D. Lanasri, "Smart-Hiring: An explainable NLP pipeline for CV information extraction and job matching," *arXiv:2511.02537*, 2025.

[13] C. Gan *et al.*, "Application of large language models in resume screening and recruitment," *arXiv:2401.08315*, 2024.

[14] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in ML-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York: Springer, 2009.

[17] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ: Wiley, 2013.

[18] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[20] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Proc. ECML PKDD*, 2008, pp. 313–325.

[21] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69–73.

[22] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst. Man Cybern. B*, vol. 39, no. 6, pp. 1362–1381, 2009.

[23] A. M. Jensi and S. Jeyabharathi, "Enhanced particle swarm optimization with Lévy flight for global optimization," *Appl. Soft Comput.*, vol. 43, pp. 248–261, 2016.

[24] C. Qin *et al.*, "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach," *arXiv:1812.08947*, 2018.

[25] Z. Wang, W. Wei, C. Xu, J. Xu, and X.-L. Mao, "Person-job fit estimation with co-attention neural networks," *arXiv:2206.09116*, 2022.

[26] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 785–794.

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ: Wiley, 2006.

[28] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, no. 6, p. 066138, 2004.

[29] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[30] D. He, X. Yuan, and H. Li, "AI and talent acquisition: The emerging frontier," *J. Bus. Res.*, vol. 112, pp. 140–148, 2020.

[31] H. Heidari, A. Ferrario, and M. B. Zafar, "Fairness in machine learning: From statistical to causal definitions," *Data Min. Knowl. Discov.*, vol. 34, no. 2, pp. 453–495, 2020.

[32] T. Zimmermann, L. Kotschenreuther, and K. Schmidt, "Data-driven HR – Résumé analysis based on NLP and machine learning," *arXiv:1606.05611*, 2016.

[33] T. T. Nguyen and T. H. Cao, "Job prediction: From deep neural network models to applications," *arXiv:1912.12214*, 2019.

[34] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, 2nd ed. Berlin: Springer, 2012, pp. 421–436.

[35] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*, 2016.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[37] O. Troyanskaya *et al.*, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, suppl. 1, pp. S412–S419, 2001.

[38] G. E. Batista and M. C. Monard, "A study of K-nearest neighbour as an imputation method," in *Proc. Brazilian Symp. Intell. Syst.*, 2003, pp. 251–260.

[39] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[40] J. W. Tukey, *Exploratory Data Analysis*. Reading, MA: Addison-Wesley, 1977.

[41] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6638–6648.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[43] A. Raghavan, S. Mohan, and P. Stone, "Bias in automated hiring systems: A systematic review," *AI Ethics*, vol. 2, no. 3, pp. 421–439, 2022.

[44] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[47] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[49] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[50] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4765–4774.

[51] S. Ö. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," in *Proc. AAAI*, vol. 35, no. 8, 2021, pp. 6679–6687.

[52] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Phys. Rev. E*, vol. 49, no. 5, pp. 4677–4683, 1994.

[53] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Off. J. Eur. Union*, L series, 2024.