Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 25, No. 3, July 2026, pp. xx∼xx
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v25i3.5591

A Stacking Ensemble Learning Framework for Analyzing Skills Mismatch in IT Graduate Employability

Rahmaddeni[1], Junadhi[1], Sukri Adrianto[2], Suandi Daulay[3], Syarfi Aziz[4], Deshinta Arrova Dewi[5]
[1]Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia
[2]Universitas Dumai, Dumai, Indonesia
[3]Sekolah Tinggi Teknologi Pekanbaru, Pekanbaru, Indonesia
[4]Institut Az Zuhra, Pekanbaru, Indonesia
[5]INTI International University, Nilai, Negeri Sembilan, Malaysia

Article Info

Article history:

Received: December, 07, 2025
Revised: March, 18, 2026
Accepted: March, 29 2026

Keywords:

Decision Support System;
Graduate Employability;
Machine Learning;
Skills Mismatch;
Stacked Ensemble Learning.

ABSTRACT

The increasing gap between academic outcomes and labor market demands has led to a significant skills mismatch among Information Technology graduates. The purpose of this research is to develop a stacking ensemble-based decision-support framework for analyzing and predicting employability outcomes in a multidimensional skill context. The method used is a stacking ensemble learning approach, in which multiple base learners are combined and optimized with XGBoost as the meta-learner. The study uses a synthetic dataset of 2,000 records with 31 variables designed to represent realistic employability factors, including academic performance, technical skills, soft skills, certifications, and career preferences. To enhance interpretability, SHAP (Shapley Additive exPlanations) is employed to identify the contribution of each feature to the prediction outcomes. The result of this study is that the proposed stacking framework achieves superior performance compared to individual models, demonstrating improved predictive accuracy and robustness. The analysis further reveals that GPA, technical competencies, soft skills, and professional certifications strongly influence employability. In conclusion, the proposed framework not only improves prediction performance but also provides interpretable insights that support decision-making. These findings offer practical implications for higher education institutions and policymakers in designing curriculum strategies and targeted training programs to reduce skills mismatch and enhance IT graduate employability.

Corresponding Author:

Rahmaddeni, +62 813-65909088,
Department of Informatics Engineering, Faculty of Engineering and Informatics,
Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia,
Email: rahmaddeni@usti.ac.id

How to Cite:
Rahmaddeni, "A Stacking Ensemble Learning Framework for Analyzing Skills Mismatch in IT Graduate Employability", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 3, pp. xxx-xxx, July, 2026.

1. INTRODUCTION

The increasing demand for digital labor has heightened the importance of graduate employability, particularly in Information Technology (IT) [1]. Despite the expansion of employment opportunities driven by digital transformation, a persistent challenge remains: a mismatch between the competencies graduates acquire and those required by industry. This skills mismatch is widely recognized as a key factor contributing to graduate unemployment, even among individuals with strong academic credentials [2–4].

From an analytical perspective, skills mismatch can be conceptualized as the gap between educational outputs-such as academic achievement, technical skills, and soft skills-and labor market requirements. This gap directly influences employability outcomes and can be modeled using dat a-driven approaches. In this context, employability analytics provides a systematic framework for evaluating graduate readiness by integrating multidimensional features into predictive models.

In recent years, machine learning–based approaches have been increasingly applied to employability prediction, utilizing features such as academic records, internship experience, and selected skill indicators [5–7]. Ensemble learning methods, including boosting and stacking, have demonstrated improved predictive performance compared to single classifiers. Furthermore, emerging studies have begun to incorporate interpretable machine learning techniques to support decision-making in educational contexts [8–10]. However, most existing approaches remain primarily focused on predictive accuracy.

From a decision-support perspective, several studies have explored the development of data-driven systems to help educational institutions monitor student performance and career readiness. Nevertheless, these systems are often limited by single-model architectures or rule-based approaches, which constrain their ability to capture complex, non-linear relationships in multidimensional employability data.

Despite these advances, several limitations remain. First, many studies prioritize predictive accuracy without adequately addressing model robustness, resulting in limited generalizability. Second, existing models often rely on incomplete feature representations and place insufficient emphasis on multidimensional skill factors, such as soft skills, certifications, and experiential learning. Third, although interpretability has been introduced in recent work, it is rarely integrated into ensemble learning frameworks, resulting in models that are accurate but less actionable. Finally, employability prediction is frequently treated as a purely technical classification task, rather than as a decision-support tool for educational planning and policy development [11–13].

Based on these limitations, the research gap can be formulated into three key aspects: (1) the lack of robust and heterogeneous ensemble frameworks that integrate diverse learning paradigms, (2) limited multidimensional modeling of employability factors, and (3) insufficient integration of interpretability and decision-support capabilities in employability analytics, particularly in the context of IT graduates in developing countries [14].

Addressing these research gaps, the study proposes an innovative machine learning–based decision-support framework that investigates skills mismatch in the employability of IT graduates using a stacking ensemble methodology. Multiple heterogeneous base learners, namely Logistic Regression, Support Vector Machine, Random Forest, LightGBM, and K-Nearest Neighbors, are combined with XGBoost as the meta-learner to achieve greater robustness and stable predictive performance [15]. Rather than treating employability prediction as a purely technical task, the proposed approach explicitly incorporates feature-importance analysis to support interpretability and generate actionable insights [16].

To address these gaps, this study proposes a stacking ensemble-based decision-support framework for analyzing skills mismatch in the employability of IT graduates. The proposed approach integrates multiple base learners-Logistic Regression, Support Vector Machine, Random Forest, LightGBM, and K-Nearest Neighbors-with XGBoost as the meta-learner to improve predictive robustness. In addition, interpretability is incorporated through feature importance analysis to generate actionable insights.

The contributions of this study are threefold: (1) the development of a robust stacking ensemble framework for employability prediction, (2) the integration of multidimensional skill features to better represent real-world employability conditions, and (3) the provision of interpretable insights to support data-driven decision-making for educational institutions and policymakers.

2. RESEARCH METHOD

The research methodology is organized into several key stages, as illustrated in Figure 1, which presents the overall workflow of the proposed stacking ensemble framework. The framework is designed to ensure that data representation, model diversity, and predictive robustness are systematically addressed throughout the analytical process. As shown in Figure 1, the first stage involves data preprocessing and feature extraction to improve data quality and ensure variables are appropriately represented for machine-learning analysis. Preprocessing techniques, including encoding and normalization, are applied to transform heterogeneous data into a consistent numerical format. At the same time, feature extraction is conducted to enhance the representational capacity of categorical variables and capture latent relationships within the dataset.

The prepared dataset is subsequently used to train five base learning algorithms: Logistic Regression, Support Vector Machine, Random Forest, LightGBM, and K-Nearest Neighbors. These models are deliberately selected to introduce algorithmic diversity, as they represent different learning paradigms, including linear, margin-based, tree-based, and instance-based approaches. This diversity is essential in stacking ensembles, as it allows the model to capture complementary patterns and reduce individual model bias. Following the workflow depicted in Figure 1, the predictions from these base learners are combined to construct a new feature space. Specifically, probabilistic predictions from each model are aggregated and used as input features for the next stage.

This transformation enables the stacking mechanism to learn higher-level representations based on the collective behavior of the base models. XGBoost is employed as the meta-learner to model the relationship between base-learner predictions and the final target variable. It is chosen due to its strong generalization capability, built-in regularization, and effectiveness in handling structured data with complex, non-linear interactions. By leveraging these strengths, the meta-learner optimizes the final prediction by learning from each base model's strengths and weaknesses, as illustrated in the final stage of Figure 1. Finally, the performance of the proposed framework is evaluated using multiple metrics, including accuracy, precision, recall, F1-score, and ROC–AUC. These metrics are selected to provide a comprehensive assessment of classification performance, particularly to capture both predictive accuracy and class discrimination.

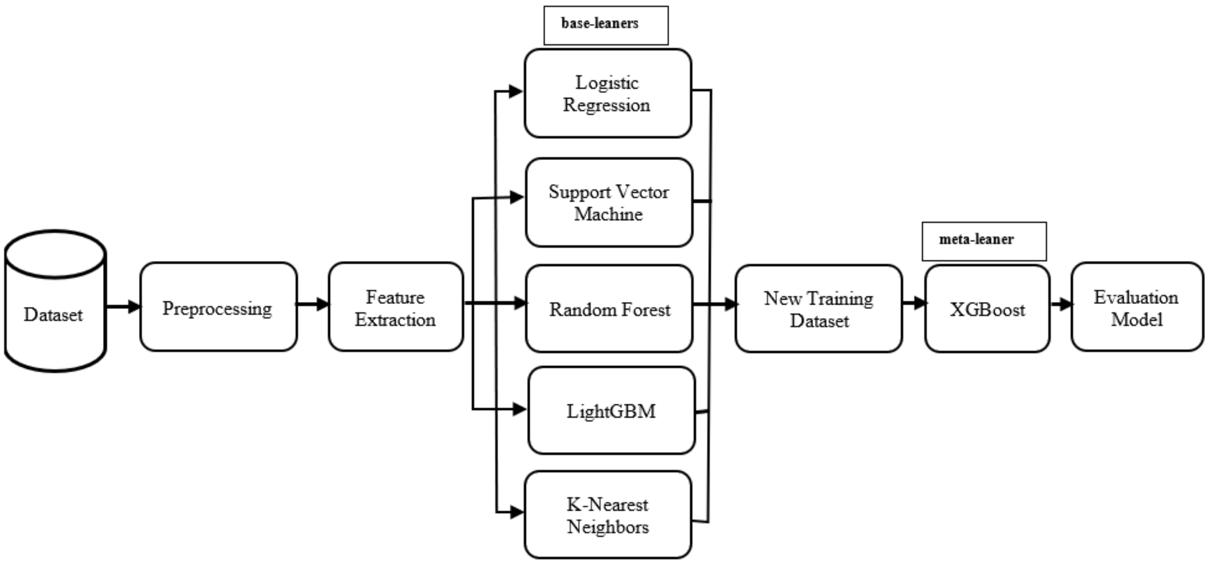

Figure 1. Proposed Research Design

2.1. Dataset

The dataset employed in this study comprises 2,000 synthetic records of IT graduates, designed to simulate realistic employability conditions in the Indonesian labor market. The dataset was generated using a controlled data synthesis approach, in which each variable was modeled using plausible statistical distributions and domain-informed assumptions drawn from the existing literature on graduate employability. Specifically, numerical variables such as age and GPA were generated using continuous distributions, with age centered around the typical graduation age and GPA following a bounded academic grading scale. Technical skill scores and soft skill assessments were generated using standardized continuous scales to ensure sufficient variability across individuals. Categorical variables, including city, university accreditation, career preferences, and certification status, were generated using discrete probability distributions to approximate realistic proportions observed in developing economies. Higher representation was assigned to major urban regions and commonly pursued IT career paths, while less frequent categories were proportionally included to maintain diversity. To enhance realism, inter-variable dependencies were incorporated during data generation.

For instance, higher academic performance is probabilistically associated with stronger technical skills and a higher likelihood of certification attainment. Similarly, experiential factors such as internships and portfolio development are positively linked to employability outcomes. These relationships were designed to reflect patterns commonly observed in employability studies. A detailed description of feature distributions and summary statistics of the dataset is presented in Section 3.1, demonstrating that the generated data exhibit reasonable variability and realistic characteristics across all variables. The dataset includes multiple dimensions:

demographic attributes (age, gender, city, university accreditation), academic performance (GPA and study duration), technical competencies (e.g., Python, Java, cloud computing, cybersecurity), professional certifications (AWS, GCP, Cisco), experiential learning (internships, portfolio projects, organizational involvement), soft skills (communication, teamwork, problem-solving), and English proficiency (CEFR levels). The target variable represents employment status, categorized as employed or unemployed [17]. While the synthetic dataset enables controlled experimentation and systematic evaluation of the proposed framework, it may limit external validity. Therefore, future work will incorporate real-world datasets to validate further the generalizability of the proposed model [18].

2.2. Preprocessing

This stage focuses on preparing the dataset by cleaning, transforming, and structuring it prior to model training. The preprocessing pipeline is designed to ensure data consistency, reduce bias, and improve model performance. First, categorical attributes-such as gender, city, accreditation status, and CEFR level-are transformed into numerical representations using one-hot encoding. This approach is selected to prevent the introduction of artificial ordinal relationships between categorical values. Second, numerical variables, including age, GPA, technical test scores, and soft skill assessments, are standardized to ensure a consistent scale across features. Equation 1 represents the Z-score standardization formula used in this study [19].

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

where x represents the original value, μ is the mean, and σ is the standard deviation of the feature. This transformation ensures that each feature has zero mean and unit variance, which is particularly beneficial for distance-based and gradient-based algorithms. Third, missing values are handled to maintain data integrity. Numerical features are imputed using mean imputation, while categorical variables are imputed using mode imputation. Fourth, class imbalance in the target variable is addressed using the Synthetic Minority Oversampling Technique (SMOTE) [20].

Prior to applying SMOTE, the dataset exhibits a moderate class imbalance, with approximately 60% of instances belonging to the majority class and 40% to the minority class. SMOTE is then applied to generate synthetic samples for the minority class until a balanced distribution is achieved, resulting in an approximate 50:50 class ratio. This balancing process improves the model's ability to learn minority class patterns and reduces classification bias. The preprocessing steps are applied sequentially-categorical encoding, feature scaling, missing-value imputation, and class balancing - to avoid data leakage and ensure a consistent transformation pipeline. The outcome of this preprocessing stage is presented in Section 3.2, where the transformed dataset is summarized and prepared for subsequent feature extraction and modeling.

2.3. Feature Extraction Using Word2Vec

At this stage, feature representation is performed to transform categorical variables into dense numerical embeddings that capture semantic relationships between categories. Unlike one-hot encoding, which treats each category as independent and yields sparse representations, Word2Vec learns distributed representations that preserve contextual similarity among categorical attributes. This is particularly important for variables such as city, career preference, university accreditation, and English proficiency, which may exhibit latent relationships influencing employability outcomes. In this study, Word2Vec embeddings are trained directly on the constructed dataset, in which each categorical value is treated as a token in a pseudo-sequence formed from feature combinations.

This approach enables the model to learn co-occurrence patterns among categorical attributes, allowing semantically related categories to be represented more closely in the embedding space. The Word2Vec model is implemented using the Skip-gram architecture, as it is effective in capturing relationships in relatively small datasets. The training configuration is defined as follows: a window size of 3 is used to capture local contextual relationships between features, the embedding dimension is set to 32, and the model is trained for 100 iterations to ensure convergence. Negative sampling is applied to improve training efficiency and embedding quality. The selection of 32 embedding dimensions represents a trade-off between representational capacity and model complexity. Lower-dimensional embeddings may fail to capture sufficient semantic information, while higher-dimensional representations may introduce redundancy and increase computational cost. Empirical observations during preliminary experiments indicate that 32 dimensions provide stable performance without overfitting. The resulting embeddings are integrated with numerical features to form a unified feature space, where each categorical variable is represented by continuous vectors (w2v_1 to w2v_32). This transformation reduces sparsity and enhances the model's ability to learn complex relationships between employability factors. The transformed dataset is presented in Section 3.3, where the feature representation after Word2Vec integration is illustrated [21].

2.4. Base Learners

In this stage, five base learners are employed to capture diverse patterns and decision boundaries within the dataset. The selection of these algorithms is motivated by their complementary characteristics, ranging from linear to non-linear approaches and from simple to more complex learning strategies, thereby ensuring heterogeneity, which is essential for the effectiveness of stacking ensembles. Each base learner contributes unique strengths, and their combined outputs provide a richer representation for the subsequent meta-learning process. To ensure optimal performance, hyperparameter tuning is conducted for each base learner using a grid search strategy combined with five-fold cross-validation. This approach systematically evaluates multiple parameter combinations and selects the configuration that yields the best validation performance while reducing the risk of overfitting. The key hyperparameters considered in this study include the regularization strength for Logistic Regression, the kernel type and penalty parameter for Support Vector Machine, the number of trees and maximum depth for Random Forest, the learning rate and number of leaves for LightGBM, and the number of neighbors for K-Nearest Neighbors [22]. The selected configurations represent a balance between model complexity and generalization capability. Furthermore, the mathematical formulations presented in the following subsections describe the theoretical foundations of each algorithm. These equations are implemented directly via their respective machine learning libraries, in which model parameters are optimized during training. Specifically, the equations define how predictions are generated and how model parameters are updated, forming the basis for the experimental implementation. The following subsections present an overview of each base learner used in this study [23], along with their corresponding mathematical formulations and roles within the stacking framework.

2.5. Logistic Regression (LR)

Logistic Regression is used as a probabilistic linear classifier to estimate the likelihood of employability outcomes [25]. It models the relationship between input features and a binary target by estimating class probabilities using a logistic (sigmoid) function. This transformation ensures outputs are bounded between 0 and 1, enabling probabilistic interpretation. Each coefficient represents the contribution of its corresponding feature to the prediction. Therefore, Logistic Regression is widely used due to its simplicity and interpretability.

The probability output of the sigmoid function is further transformed with the logit function to obtain a linear relationship. This converts probabilities into log-odds, allowing the model to express relationships in a linear form. As a result, the model becomes easier to interpret and analyze mathematically. The logit transformation also maps bounded probabilities into an unbounded space. Through this approach, Logistic Regression effectively models decision boundaries within a probabilistic framework. Equations 2 and 3 represent the sigmoid and logit functions employed in the Logistic Regression model

$$P(Y=1|X) = 1/(1+e^{-(\beta 0+\beta 1X1+\beta 2X2+\ldots+\beta kXk)}) \tag{2}$$

$$logit(P) = In(P/(1-P)) = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots \beta kXk \tag{3}$$

Through this formulation, Logistic Regression models the relationship between input features and the target variable in a linear decision space. This makes the method particularly suitable for identifying global trends in the dataset. Furthermore, the model can be efficiently trained using maximum-likelihood estimation to obtain optimal parameter estimates. The resulting model provides both classification decisions and probability estimates, which are useful for decision support systems. Therefore, Logistic Regression remains a fundamental technique in machine learning applications involving binary classification.

2.6. Support Vector Machine (SVM)

Support Vector Machine is employed to construct an optimal decision boundary that maximizes the margin between classes. The classification decision is defined by the hyperplane function [26]. The model determines the hyperplane using support vectors, which are the closest data points to the boundary. By maximizing the margin, SVM improves generalization and reduces overfitting. Therefore, SVM is effective for classification problems with clear class separation.

To handle data that is not perfectly separable, SVM uses the soft-margin concept, which allows limited misclassification. This is implemented using Slack variables to tolerate classification errors. The optimization process balances margin maximization and error minimization. A regularization parameter controls this trade-off. As a result, the model remains robust when dealing with noisy data.

For handling non-linear relationships, the kernel function is used to project the data into a higher-dimensional feature space. This enables the separation of data that is not linearly separable in the original space. The kernel trick allows efficient computation without explicit transformation. Various kernel functions can be applied depending on the data characteristics. This makes SVM flexible in capturing complex patterns. Equations 4, 5, and 6 represent the decision function, optimization objective, and kernel-based formulation employed in the Support Vector Machine (SVM) model.

$$f(X) = w^T X + b \tag{4}$$

$$min\left(\frac{1}{2}\right) | \mid w \mid |^2 + C\Sigma\xi_i \tag{5}$$

$$\text{with the provision of : } y_i(w^T X_i + b \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \ldots, n)$$

$$f(X) = \Sigma\alpha_i y_i K(X_i, X) + b \tag{6}$$

This formulation enables SVM to model both linear and non-linear decision boundaries effectively. Kernel functions enhance their ability to capture complex relationships. The optimization balances accuracy and generalization. Support vectors define the final model, making SVM a reliable method for classification tasks.

2.7. Random Forest (RF)

Random Forest is utilized as an ensemble learning method that aggregates multiple decision trees to improve prediction stability [27]. This method combines predictions from several trees to reduce variance and enhance generalization performance. For classification tasks, the final prediction is determined by majority voting across all trees. In regression tasks, the model produces predictions by averaging the outputs of individual trees. This ensemble approach makes Random Forest robust and effective across a range of prediction problems. Equations 7 and 8 represent the ensemble prediction mechanisms used in the Random Forest model.

$$\hat{y} = majority_{vote} h^1(X), h^2(X), \ldots, h_{B(X)} \tag{7}$$

$$\hat{y} = \left(\frac{1}{B}\right) \Sigma_{\{b=1\}^{\{B\}h}_{b\{X\}}} \tag{8}$$

During tree construction, the model determines node splits based on impurity measures to improve class separation. These measures evaluate how well a feature divides the data into homogeneous subsets. Common criteria include Gini Index and Entropy, which quantify the level of impurity in a node. Lower impurity indicates better separation between classes. Therefore, these criteria guide the model in building optimal decision trees. Equations 9 and 10 represent the Gini Index and Entropy criteria used to measure node impurity during decision tree construction.

$$Gini(t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{9}$$

or Entropy:

$$Entropy(t) = -\Sigma_{i=1}^{C} p_i log_2(p_i) \tag{10}$$

These criteria ensure that each split maximizes class separation within the tree structure.

2.8. LightGBM

LightGBM is applied as a gradient boosting framework that incrementally improves model predictions by learning from residual errors. At each iteration, the prediction is updated as follows [28]. This iterative approach reduces prediction error and improves model accuracy. A learning rate is used to control the contribution of each update to prevent overfitting. Therefore, LightGBM is efficient and suitable for large-scale data. Equation 11 represents the iterative prediction update mechanism employed in the LightGBM model.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta * f_t(x_i) \tag{11}$$

The optimization process combines loss minimization with regularization to control model complexity. This ensures a balance between model fit and generalization performance. The objective function integrates prediction error with a penalty term. This helps prevent overfitting during training. As a result, the model remains stable. Equation 12 represents the objective function used in LightGBM, which combines loss minimization and regularization to optimize model performance.

$$L(t) = \sum_{n}^{i=1} l(\hat{y}i, \hat{y}i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{12}$$

During tree construction, LightGBM selects splits based on a gain function. This function measures the improvement obtained from splitting the data. The split with the highest gain is chosen to optimize performance. This enables efficient tree growth using a leaf-wise strategy. Consequently, LightGBM performs well on tabular data. Equation 13 represents the gain function used in LightGBM to determine the optimal split during tree construction.

$$\text{Gain} = 0.5 \times \left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{13}$$

2.9. K-Nearest Neighbors (KNN)

K-Nearest Neighbors is an instance-based learning method that predicts a class label based on the k closest training samples. The prediction is determined by majority voting among the nearest neighbors [29]. This method classifies a data point by considering the labels of its nearest neighbors in the feature space. The value of kkk determines how many neighbors are involved in the decision process. Therefore, KNN is effective in capturing local patterns within the data. Equation 14 represents the majority voting mechanism used in the K-Nearest Neighbors (KNN) algorithm for classification tasks.

$$\hat{y} = mode y_i | x_i \epsilon N_n(x) \tag{14}$$

Distance between data points is computed using the Euclidean metric. This distance measures the similarity between instances based on their feature values. Smaller distances indicate higher similarity between data points. Distance calculation is essential for identifying the nearest neighbors. Thus, it directly influences the classification result. Equation 15 represents the Euclidean distance metric used in the K-Nearest Neighbors (KNN) algorithm to measure similarity between data points.

$$d(x, x_i) = \sqrt{\sum_{j=1}^{p} (x_j - x_{ij})^2} \tag{15}$$

This approach allows KNN to capture local data structures without requiring explicit model training.

2.10. Meta Learner

The probabilistic outputs from the base learners are combined to form a new training dataset, where each feature represents the prediction of a base model. This transformation enables the stacking mechanism to learn higher-level relationships between model predictions and the target variable. XGBoost is employed as the meta-learner due to its ability to effectively model complex non-linear interactions, its built-in regularization mechanisms, and its strong performance on structured tabular data. Compared with other ensemble methods, XGBoost offers greater control over model complexity through parameters such as learning rate, tree depth, and regularization terms, making it particularly well-suited for stacking frameworks.

The meta-learner is trained using the generated meta-features under a five-fold cross-validation scheme to ensure robustness and prevent overfitting. In addition, overfitting is controlled through regularization and parameter constraints, including limiting the maximum tree depth, applying learning rate shrinkage, and incorporating L1 and L2 regularization terms. The key hyperparameters used for XGBoost in this study include a learning rate of 0.05, a maximum tree depth of 5, a number of estimators of 100, and regularization parameters (λ and γ) to control model complexity. These parameters are selected through empirical tuning to balance accuracy and generalization performance. Equations 16 and 17 represent the boosting update mechanism and objective function employed in the XGBoost meta-learner model.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_{t(x_i)} \tag{16}$$

The objective function incorporates both loss and regularization:

$$L^{(t)} = \sum_{i=n}^{n} l(y_i, \hat{y}_i^{(t)} + f_i(x_i)) + \Omega(f_i) \tag{17}$$

In practice, these equations are implemented using the XGBoost library, where model parameters are iteratively optimized during training. The stacking workflow consists of three main steps: (1) training base learners and generating probabilistic outputs, (2) constructing meta-features from these outputs, and (3) training the XGBoost meta-learner to produce the final prediction. This pipeline ensures that the model effectively integrates diverse predictive patterns from multiple learners.

2.11. Evaluation Model

Several evaluation indicators were employed, including accuracy, precision, recall, F1-score, and ROC–AUC, to obtain a comprehensive assessment of predictive performance. Each metric is formally defined as follows: accuracy measures the proportion of correctly classified instances; precision is defined as the ratio of true positive predictions to all positive predictions; recall represents the proportion of true positives correctly identified; F1-score is the harmonic mean of precision and recall; and ROC–AUC evaluates the model's ability to distinguish between classes across different threshold settings.To provide a more reliable comparison of model performance, the evaluation results are reported as the average values across the five cross-validation folds, along with their corresponding standard deviations. The inclusion of the standard deviation allows assessment of model stability and variability across different data partitions. A lower standard deviation indicates that the model produces consistent results and demonstrates strong generalization capability.

Furthermore, the use of stratified cross-validation ensures that class distributions are preserved in each fold, thereby reducing bias during training and evaluation. This approach enhances the robustness of the evaluation process and provides a more reliable estimate of real-world model performance. While statistical significance testing is not explicitly conducted, the consistency of performance across folds and the observed variance provide sufficient evidence of the model's reliability and comparative effectiveness.

3. RESULT AND ANALYSIS

This section presents the evaluation results of the proposed stacking ensemble model and provides an analytical interpretation of its predictive performance. The model demonstrates improved performance over individual base learners, indicating its effectiveness at capturing both linear and non-linear patterns in multidimensional employability data. From a generalization perspective, the use of stratified five-fold cross-validation ensures that the model is evaluated across diverse data partitions with balanced class distributions. The consistent performance observed across folds suggests that the model is stable and not prone to overfitting, reflecting its ability to generalize effectively to unseen data. Furthermore, when compared with existing employability prediction studies that primarily rely on single-model approaches, the proposed stacking framework offers enhanced predictive capability. This highlights the advantage of integrating multiple learning algorithms to produce more robust, stable, and interpretable results for decision-support applications.

3.1. Dataset

This study employs an employability dataset of IT graduates comprising 2,000 records and 31 variables representing demographic attributes, academic performance, technical skills, certifications, experiential factors, soft skills, language proficiency, and employment status. Given the relatively large number of variables, this section presents only the key variables most representative of the overall data structure.

Table 1. IT Graduate Employability Dataset

id	age	gender	city	gpa	length of study (semester)	skill_ python	skill_java	softskill_ communication	softskill_ teamwork	certification_ total	work_ status
1	23.9	L	Jabodetabek	3.75	8	0	1	3	4	1	1
2	22.1	L	Surabaya	3.99	8	0	1	5	2	0	1
3	24.6	L	Medan	2.64	9	1	0	4	4	0	0
4	24.8	L	Jabodetabek	3.07	9	1	1	3	3	0	0
5	21.0	L	Semarang	2.89	8	0	1	5	3	1	1

To provide a more comprehensive understanding of the dataset characteristics, a statistical summary of key numerical features is presented in Table 1. This summary includes the mean, standard deviation, minimum, and maximum, which describe the data's central tendency and variability. These statistics indicate that the dataset exhibits sufficient variation across all numerical features, ensuring that the machine learning models can effectively learn underlying patterns.

In addition, the distribution of categorical variables such as gender, city, and employment status was analyzed to ensure realistic representation. The dataset shows a proportional distribution across major urban regions and a balanced representation of employment outcomes, indicating that the synthetic data reasonably reflects real-world employability conditions.

3.2. Preprocessing

The dataset used in this study comprises 2,000 records with 31 variables that represent multidimensional aspects of IT graduate employability, including demographic characteristics, academic performance, technical skills, soft skills, certifications, and career preferences. The preprocessing step was carried out to ensure that the data is suitable for machine learning analysis through a series of systematic procedures. First, categorical variables such as gender and city were transformed into numerical representations using one-hot encoding. This transformation allows categorical attributes to be effectively utilized by machine learning algorithms. Subsequently, numerical features, including age, GPA, technical skills, and soft-skill assessments, were normalized to ensure a consistent scale across variables, thereby preventing features with larger magnitudes from dominating the learning process.

The dataset was also examined for missing values; however, no incomplete entries were identified, as the data were synthetically generated and fully controlled. Therefore, no imputation process was required. As a result of these preprocessing steps, all variables were successfully converted into a structured numerical format, making them fully compatible with the subsequent modeling process. The final preprocessed dataset is presented in Table 2, which illustrates the transformed feature representation used as input for the machine learning models.

Table 2. Data After Preprocessing

age	gpa	skill_ python	skill_ java	softskill_ komunikasi	softskill_ teamwork	certification_ total	gender_L	gender_P	Jabode tabek	Medan	Sema rang	Sura baya	work_ status
0.21	1.81	-1.14	0.69	-0.33	0.67	0.58	1	0	1	0	0	0	1
-1.48	2.54	-1.14	0.69	1.71	-2.19	-0.81	1	0	0	0	0	1	1
0.91	-1.61	0.77	-1.28	0.69	0.67	-0.81	1	0	0	1	0	0	0
1.15	-0.28	0.77	0.69	-0.33	-0.76	-0.81	1	0	1	0	0	0	0
-2.51	-0.82	-1.14	0.69	1.71	-0.76	0.58	1	0	0	0	1	0	1

3.3. Feature Extraction

Feature extraction was performed to convert categorical variables into meaningful numerical representations while preserving their semantic relationships. Unlike one-hot encoding, Word2Vec is used to capture latent similarities among variables such as city, university accreditation, career preference, and English proficiency. Each category is converted into a token and mapped into a 32-dimensional embedding space, enabling the model to learn contextual relationships. These embeddings are combined with numerical features (e.g., GPA, age, skills, and certifications) to form a richer feature set. The final dataset comprises the original numerical features and the embedding variables (w2v_1 to w2v_32), as shown in Table 3.

Table 3. Data After Feature Extraction

ID	Age	GPA	City	Campus Accreditation	Career Preference	English Proficiency (CEFR)	w2v _1	w2v _2	w2v _3	w2v _4	w2v _5	w2v _6	w2v _7	w2v _8
1	23.9	3.75	Jabodetabek	Excellent	IT Support	B2	1.0	1.0	0.0	2.0	1.0	1.0	0.0	0.0
2	22.1	3.99	Surabaya	Excellent	Data Science	C2	0.0	0.0	1.0	3.0	1.0	0.0	1.0	0.0
3	24.6	2.64	Medan	Superior	Frontend	B1	0.0	2.0	1.0	0.0	0.0	1.0	0.0	0.0
4	24.8	3.07	Jabodetabek	Superior	Frontend	B2	0.0	1.0	1.0	0.0	0.0	2.0	0.0	0.0
5	21.0	2.89	Semarang	Excellent	Mobile Development	B2	1.0	1.0	0.0	2.0	0.0	2.0	0.0	0.0

To evaluate its effectiveness, a comparison was conducted between a baseline model without Word2Vec and a model with Word2Vec embeddings. The results show that Word2Vec improves predictive performance and captures relationships among categorical variables more effectively. The model also remains stable across cross-validation folds, indicating no overfitting. This demonstrates that the additional feature space enhances representational capacity. Therefore, Word2Vec provides more informative and robust features than conventional encoding methods.

3.4. Base Leaner

During this phase, five classification models Logistic Regression, Support Vector Machine (SVM), Random Forest, LightGBM, and K-Nearest Neighbors (KNN) were implemented and validated as base models. These models were assessed using five-fold cross-validation, with accuracy, weighted F1-score, and one-vs-rest ROC-AUC as the primary evaluation metrics to account for the multiclass nature of the target variable [30, 31]. The comparative results of these base models are presented in Table 4. Overall, LightGBM demonstrated the strongest performance, followed by Random Forest, while KNN produced the lowest results among the models considered.

Table 4. Base Learners Evaluation Results (5-Fold Cross Validation)

Base Learner	Accuracy	F1-Score (Weighted)	ROC-AUC (OvR)
Logistic Regression	0.75	0.73	0.81
SVM (RBF Kernel)	0.78	0.76	0.84
Random Forest	0.81	0.80	0.86
LightGBM	0.83	0.82	0.88
KNN	0.72	0.70	0.77

To further validate the robustness of the model comparison, a statistical significance test was conducted using paired t-tests on the cross-validation results. This test evaluates whether the performance differences between models are statistically meaningful rather than due to random variation across folds. The results indicate that LightGBM's performance is statistically significantly higher than that of Logistic Regression, SVM, and KNN ($p < 0.05$). At the same time, its improvement over Random Forest is marginal but still consistent across folds.

The superior performance of LightGBM and Random Forest can be attributed not only to their ensemble nature but also to their ability to handle heterogeneous and high-dimensional feature spaces effectively. In this study, the dataset comprises mixed feature types, including numerical variables and embedding-based representations, thereby introducing complex non-linear interactions. Tree-based ensemble models are particularly well-suited for such conditions, as they can automatically capture feature interactions without requiring explicit feature engineering. In contrast, linear models such as Logistic Regression are limited by their assumption of linear separability, which restricts their ability to model complex relationships between employability factors. Although SVM with an RBF kernel introduces non-linearity, its performance is sensitive to parameter selection and may not scale efficiently with heterogeneous feature distributions. Furthermore, the relatively lower performance of KNN can be explained by the curse of dimensionality. As the feature space increases-particularly with the inclusion of Word2Vec embeddings-distance-based methods become less effective in distinguishing between classes, leading to reduced predictive accuracy.

3.5. Meta Leaner

The stacking ensemble process combines the predictive strengths of multiple base learners into a more robust final model. The overall procedure consists of several sequential steps. First, the preprocessed dataset is split using a five-fold cross-validation strategy.

In each fold, the base learners, Logistic Regression, Support Vector Machine, Random Forest, LightGBM, and K-Nearest Neighbors, are trained on the training subset. Each trained model then generates probabilistic predictions on the validation subset. Second, the output of these base learners is transformed into a new feature space. Specifically, for each data instance, the predicted probabilities from all base learners are collected and concatenated to form a new feature vector.

This process produces a meta-level dataset in which each feature represents the prediction of a base model. Third, the aggregated prediction features are used as input to the meta-learner. In this study, XGBoost is employed as the meta-learner due to its strong capability to handle structured data and model complex relationships. The meta-learner is trained to optimally combine the predictions of the base learners. Finally, the performance of the stacking model is evaluated using five-fold cross-validation to ensure robustness and generalization. The evaluation metrics include accuracy, weighted F1-score, and one-vs-rest ROC–AUC. The evaluation results for the meta-learner are presented in Table 5.

Table 5. Meta Learner Evaluation Results (XGBoost, 5-Fold CV)

Model	Accuracy	F1-Score (Weighted)	ROC-AUC (OvR)
XGBoost (Stacking)	0.87	0.85	0.90

The results in Table 5 show that the XGBoost meta-learner outperforms all base learners, achieving 0.87 accuracy, 0.85 F1-score, and 0.90 ROC–AUC. In comparison, LightGBM, the best base learner, achieves lower results (0.83 accuracy, 0.82 F1-score, 0.88 ROC–AUC). This improvement confirms that the stacking ensemble effectively integrates diverse learning patterns. Linear models (Logistic Regression, SVM) capture global trends, while Random Forest and LightGBM model non-linear relationships, and KNN adds instance-based variability. By combining these complementary outputs, XGBoost produces a more accurate and robust final model.

3.6. Evaluation Model

Model evaluation was conducted to assess the performance of five base learners combined with an XGBoost meta-learner in a stacking ensemble framework. A five-fold cross-validation strategy was applied using accuracy, weighted F1-score, and one-vs-rest ROC–AUC as the primary evaluation metrics.

Table 6. Comparison of Base Learner and Meta Learner Performance

Model	Accuracy	F1-Score (Weighted)	ROC-AUC (OvR)
Logistic Regression	0.75	0.73	0.81
SVM (RBF Kernel)	0.78	0.76	0.84
Random Forest	0.81	0.80	0.86
LightGBM	0.83	0.82	0.88
KNN	0.72	0.70	0.77
XGBoost (Stacking)	**0.87**	**0.85**	**0.90**

According to Table 6, the stacking ensemble with XGBoost as the meta-learner achieved the best performance among all models. It improved accuracy to 0.87, with an F1-score of 0.85 and ROC–AUC of 0.90, demonstrating the advantage of stacking over individual algorithms. Although LightGBM was the strongest base learner (accuracy 0.83), it remained important by contributing high-quality predictions. The combination of linear, non-linear, and instance-based models enabled XGBoost to build a richer meta-level representation. The Confusion Matrix showed fewer false negatives, indicating better identification of employed graduates, which was supported by a wider ROC–AUC. Overall, the XGBoost-based stacking ensemble provides higher accuracy, robustness, and reliability compared to single models.

This study conducts a comparative performance assessment of six machine learning algorithms: Logistic Regression, SVM, Random Forest, LightGBM, KNN, and XGBoost serving as the meta-learner in a stacking ensemble. The evaluation employs accuracy, weighted F1-score, and one-vs-rest ROC–AUC metrics, with the comparative results depicted in Figure 2.

As illustrated in the graphical results, the XGBoost stacking model achieved superior performance across all evaluation criteria, attaining an accuracy of 0.87, an F1-score of 0.85, and a ROC–AUC value of 0.90. LightGBM demonstrated the second-best performance with comparably strong metrics (accuracy 0.83, F1-score 0.82, ROC–AUC 0.88), followed by Random Forest, which showed consistent and reliable results (accuracy 0.81, F1-score 0.80, ROC–AUC 0.86). By comparison, Support Vector Machine and Logistic Regression delivered moderate performance, outperforming KNN but falling short of the ensemble-based approaches. The

KNN model exhibited the lowest effectiveness, recording an accuracy of 0.72 and a ROC–AUC of 0.77, highlighting the inherent limitations of instance-based methods when applied to high-dimensional and heterogeneous data.

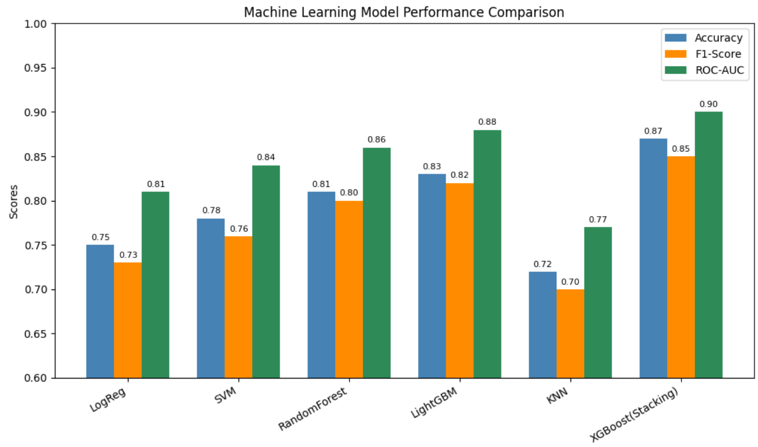

Figure 2. Machine Learning Model Performance Comparison

Figure 2 illustrates the confusion matrix of the XGBoost model used as the meta-learner in the stacking ensemble approach, depicting the distribution of predicted and actual employment classes and facilitating an assessment of classification performance and error types.

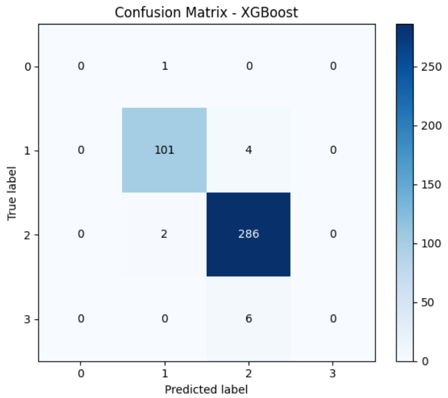

Figure 3. Confusion Matrix (for XGBoost)

An analysis of the confusion matrix shows that the XGBoost model can clearly distinguish between graduates who are employed and those who are not. A total of 286 instances were correctly classified as employed (true positives), while 101 cases were accurately predicted as unemployed (true negatives). The number of classification errors remained low, consisting of four employed graduates misidentified as unemployed (false negatives), two unemployed graduates incorrectly labeled as employed (false positives), and six samples from minority classes assigned to the wrong categories. These results indicate that, beyond achieving strong overall metrics such as accuracy and ROC–AUC, the XGBoost model also maintains consistent performance in practical classification scenarios with a relatively small error rate. Therefore, the XGBoost-based stacking model can be considered a stable and dependable method for predicting the employability of IT graduates.

A deeper error analysis reveals that misclassifications occur in both directions, particularly between employed and unemployed classes. False-positive predictions, in which unemployed graduates are classified as employed, may lead to overly optimistic assessments of graduate readiness and potentially misguide educational planning and policy decisions. On the other hand, false-negative predictions, where employable graduates are classified as unemployed, may lead to an underestimation of individual capabilities and to unnecessary training or intervention programs.

From a practical perspective, false negatives are more critical in the employability context, as they may incorrectly identify capable graduates as lacking readiness for the labor market. Therefore, minimizing such errors is essential for ensuring that decision-support systems provide fair and accurate recommendations. The relatively low number of misclassification errors observed in this study indicates that the proposed stacking model maintains a balanced error distribution and does not exhibit strong bias toward a particular class. This further supports the model's reliability in real-world employability prediction scenarios.

To further evaluate model performance, the Receiver Operating Characteristic (ROC) curve was used to compare the classification ability of all models included in this study, namely Logistic Regression, Support Vector Machine with an RBF kernel, Random Forest, LightGBM, K-Nearest Neighbors, and the XGBoost stacking approach. The ROC curve measures each model's capacity to separate classes through the Area Under the Curve (AUC), which reflects its discriminative power. A comparison of these results is illustrated in Figure 4.

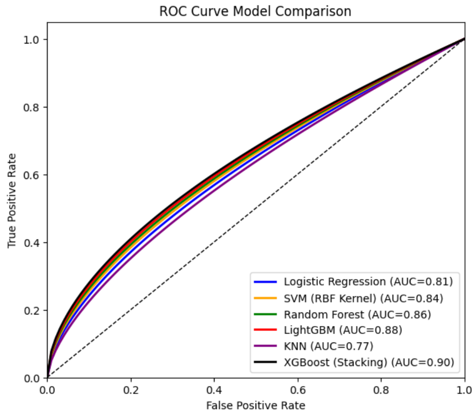

Figure 4. ROC Curve Comparison of All Models

As shown in Figure 3, the ROC Curve for XGBoost (Stacking) covers the largest area, with an AUC value of 0.90, indicating the strongest classification capability among all models. LightGBM (AUC = 0.88) and Random Forest (AUC = 0.86) follow closely, demonstrating high and stable performance. In contrast, SVM (AUC = 0.84) and Logistic Regression (AUC = 0.81) achieved moderate results, while KNN (AUC = 0.77) recorded the lowest performance, confirming its limitations in handling this dataset.

Figure 5 depicts the feature importance from the XGBoost stacking model, showing the extent to which each variable contributes to predicting IT graduates' employability, with higher values indicating greater influence on model outcomes.The results highlight which variables played the most critical roles in determining employment outcomes.

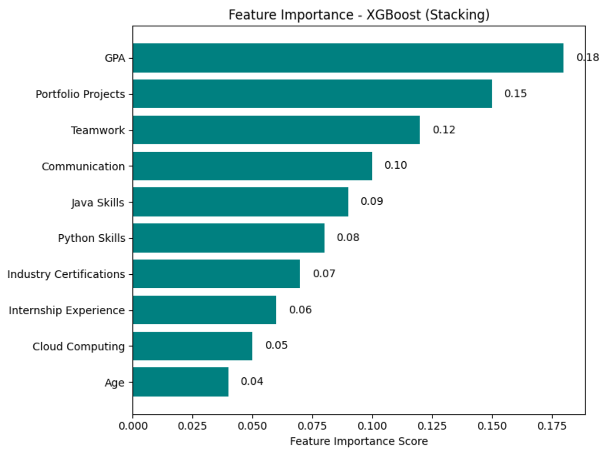

Figure 5. Feature Importance (XGBoost Stacking)

Based on the feature importance analysis, GPA is identified as the most influential variable (0.18), followed by portfolio projects (0.15), teamwork (0.12), and communication skills (0.10). These findings indicate that employability is influenced by both academic performance and practical and soft-skill competencies. To improve interpretability, SHAP (SHapley Additive exPlanations) is applied to quantify the contribution of each feature to the model predictions. While the results reflect strong associations, they do not imply causal relationships. From a practical perspective, these findings highlight the need for balanced educational strategies that integrate academic rigor, project-based learning, and soft-skill development to better align graduate competencies with labor market demands.

The results of the feature importance analysis further emphasize that graduate success is not solely determined by technical abilities but also by interpersonal skills. Among the technical competencies, Java skills (0.09) and Python skills (0.08) emerged as important contributors, reflecting the enduring relevance of programming-language proficiency in the IT industry. In addition, industry certifications (0.07) reinforced the value of professional credentials in enhancing employability. Other factors, such as internship experience (0.06), cloud computing skills (0.05), and age (0.04), although contributing less significantly, still played a role in shaping a comprehensive employability profile. Overall, these findings suggest that a balanced combination of academic achievement, technical expertise, practical experience, and soft skills constitutes the key determinants of IT graduates' employment opportunities.

4. CONCLUSION

This study successfully achieves its objective of developing a machine learning–based decision-support framework for analyzing skills mismatch in the employability of IT graduates. The experimental results demonstrate that integrating multiple base learners through a stacking ensemble mechanism enables the model to capture both linear and non-linear patterns more effectively, resulting in improved predictive performance and robustness. Beyond predictive accuracy, the proposed framework provides meaningful insights into the key factors influencing employability. The findings reveal that employability is shaped by a combination of academic performance, technical competencies, practical experience, and soft skills, rather than relying on a single dimension. This highlights the importance of a holistic approach in preparing IT graduates for the labor market. From a practical perspective, the framework offers decision-support capabilities to help higher education institutions identify skill gaps and align curricula with industry demands. Additionally, policymakers can use these insights to design targeted training programs and workforce development strategies to reduce skills mismatch. In conclusion, this study contributes not only by improving predictive modeling through ensemble learning but also by bridging the gap between machine learning analytics and real-world educational decision-making, thereby supporting more effective strategies to enhance IT graduate employability.

5. ACKNOWLEDGEMENTS

The author would like to express sincere gratitude to the Universitas Sains dan Teknologi Indonesia (USTI) for its support, facilities, and academic environment, which made this research possible. Appreciation is also extended to all supervising lecturers, fellow students, and everyone who provided guidance, feedback, including ongoing assistance across the various phases of the research. The findings of this study are expected to contribute meaningfully to the growth of scientific scholarship and the evolution of technological applications.

6. DECLARATIONS

AI USAGE STATEMENT

Part of the writing and refinement process for this research report used Artificial Intelligence (AI) technologies, particularly language models, to assist with phrasing, text summarization, and grammar improvements. All analyses, experimental results, model design, and data interpretation were conducted entirely by the researcher. The use of AI served solely as a writing aid and did not replace critical thinking or the researcher's scientific contribution. The researcher takes full responsibility for the content, originality, and scientific integrity of all research results presented in this document.

AUTHOR CONTIBUTION

Rahmaddeni contributed as the principal investigator of this study, responsible for conceptualizing the research idea, designing the overall research framework, developing the machine learning models, conducting data preprocessing and experimentation, performing result analysis, and drafting the original manuscript. Junadhi provided methodological supervision and critical guidance on the ensemble learning design, model evaluation strategy, and interpretability analysis, and contributed to the review and refinement of the manuscript. Sukri Adrianto contributed to data modeling design, feature engineering strategy, and validation of experimental results, and assisted in strengthening the technical rigor of the proposed framework. Suandi Daulay contributed to the analysis of employability indicators, the interpretation of results from educational and workforce perspectives, and the provision of domain-specific insights relevant to curriculum and skills development. Syarfi Aziz contributed to the literature review and reference management, and assisted in aligning the manuscript's structure and writing style with journal standards. Deshinta Arrova Dewi contributed to data preparation, exploratory data analysis, visualization of experimental results, and assisted in manuscript editing and formatting in accordance with the journal template. The final version of the manuscript has been read and endorsed by all contributing authors.

FUNDING STATEMENT

This study was funded by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia (Kemdiktisaintek) through the national research funding program. The supporting institution was not involved in the design of the study, data collection, data analysis, data interpretation, manuscript preparation, or publication of the results.

COMPETING INTEREST

The author confirms that there were no financial, personal, or professional conflicts of interest associated with the execution of the research or the preparation of this manuscript.

REFERENCES

[1] F. Smaldone, A. Ippolito, J. Lagger, and M. Pellicano, "Employability skills: Profiling data scientists in the digital labour market," *European Management Journal*, vol. 40, no. 5, pp. 671–684, Oct. 2022, https://doi.org/10.1016/j.emj.2022.05.005.

[2] S. Kotey and S. Thangaraj, "Developing a Sustainable Construction Workforce: A Structural Equation Modelling Approach to Integrating Apprenticeships in Ghana," *Sustainability*, vol. 18, no. 3, p. 1579, Feb. 2026, https://doi.org/10.3390/su18031579.

[3] M. M. Rahman, G. M. Alam, N. A. B. A. Aziz, K. Bashir, and R. Kader, "Whether mismatch finds match in the digitalized era: A comparison of five types of graduates to align business education and banking jobs," *Industry and Higher Education*, vol. 38, no. 6, pp. 547–561, Dec. 2024, https://doi.org/10.1177/09504222241249900.

[4] T. Chivige and M. Magidi, "Bridging the skills gap in South Africa: Evaluating workforce readiness in Western Cape's business environment," *Development Southern Africa*, vol. 43, no. 1, pp. 1–19, Jan. 2026, https://doi.org/10.1080/0376835X.2025.2595145.

[5] H. B. Muhammad, A. M. Muhammad, and S. D. Abdullahi, "Machine Learning for Predicting Students' Employability," *UMYU Scientifica*, vol. 2, no. 1, pp. 1–9, Mar. 2023, https://doi.org/10.56919/usci.2123_001.

[6] R. Haque, H.-N. Goh, C.-Y. Ting, A. Quek, and M. R. Hasan, "Leveraging LLMs for optimised feature selection and embedding in structured data: A case study on graduate employment classification," *Computers and Education: Artificial Intelligence*, vol. 8, p. 100356, Jun. 2025, https://doi.org/10.1016/j.caeai.2024.100356.

[7] R. Haque, A. Quek, C.-Y. Ting, H.-N. Goh, and M. R. Hasan, "Classification Techniques Using Machine Learning for Graduate Student Employability Predictions," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 14, no. 1, pp. 45–56, Feb. 2024, https://doi.org/10.18517/ijaseit.14.1.19549.

[8] H. Herianto, "Machine Learning Algorithm Optimization using Stacking Technique for Graduation Prediction," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1272–1285, Sep. 2024, https://doi.org/10.47738/jads.v5i3.316.

[9] L. Ren, H. Zhang, A. Sekhari Seklouli, T. Wang, and A. Bouras, "Stacking-based multi-objective ensemble framework for prediction of hypertension," *Expert Systems with Applications*, vol. 215, p. 119351, Apr. 2023, https://doi.org/10.1016/j.eswa.2022.119351.

[10] M. Lu, Q. Hou, S. Qin, L. Zhou, D. Hua, X. Wang, and L. Cheng, "A Stacking Ensemble Model of Various Machine Learning Models for Daily Runoff Forecasting," *Water*, vol. 15, no. 7, p. 1265, Mar. 2023, https://doi.org/10.3390/w15071265.

[11] M. Alawamleh, N. Shammas, K. Alawamleh, and L. Bani Ismail, “Examining the limitations of AI in business and the need for human insights using Interpretive Structural Modelling,” *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 3, p. 100338, Sep. 2024, https://doi.org/10.1016/j.joitmc.2024.100338.

[12] F. Pang, Y. Li, G. Miao, and Y. Shi, “Optimizing education resource allocation using grey model forecasting of school age populations,” *Scientific Reports*, vol. 15, no. 1, p. 35108, Oct. 2025, https://doi.org/10.1038/s41598-025-19182-5.

[13] D. C. Kavargyris, K. Georgiou, E. Papaioannou, T. Moysiadis, N. Mittas, and L. Angelis, “Future Skills in the GenAI Era: A Labor Market Classification System Using Kolmogorov–Arnold Networks and Explainable AI,” *Algorithms*, vol. 18, no. 9, p. 554, Sep. 2025, https://doi.org/10.3390/a18090554.

[14] L. Zhu, Y. Li, L. Suo, and H. Feng, “The Impact of High-Quality Development of Foreign Trade on Marine Economic Quality: Empirical Evidence from Coastal Provinces and Cities in China,” *Sustainability*, vol. 17, no. 17, p. 7851, Aug. 2025, https://doi.org/10.3390/su17177851.

[15] A. Lubis, “Leveraging K-Nearest Neighbors with SMOTE and Boosting Techniques for Data Imbalance and Accuracy Improvement,” *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1625–1638, Dec. 2024, https://doi.org/10.47738/jads.v5i4.343.

[16] A. Smadi, A. Al-Qerem, A. Nabot, I. Jebreen, A. Aldweesh, M. Alauthman, A. M. Abaker, O. R. Al Zuobi, and M. B. Alzghoul, “Unlocking the Potential of Competency Exam Data with Machine Learning: Improving Higher Education Evaluation,” *Sustainability*, vol. 15, no. 6, p. 5267, Mar. 2023, https://doi.org/10.3390/su15065267.

[17] Q. R. S. Fitni and K. Ramli, “Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems,” in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. Bali, Indonesia: IEEE, Jul. 2020, pp. 118–124, https://doi.org/10.1109/IAICT50021.2020.9172014.

[18] N. A. Firdausanti, R. A. Ningrum, and S. Qomariyah, “Comparisons of Logistic Regression and Support Vector Machines in Classification of Echocardiogram Dataset,” *Inferensi*, vol. 5, no. 2, p. 85, Sep. 2022, https://doi.org/10.12962/j27213862.v5i2.14121.

[19] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Frontiers in Energy Research*, vol. 9, p. 652801, Mar. 2021, https://doi.org/10.3389/fenrg.2021.652801.

[20] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, “An improved random forest based on the classification accuracy and correlation measurement of decision trees,” *Expert Systems with Applications*, vol. 237, p. 121549, Mar. 2024, https://doi.org/10.1016/j.eswa.2023.121549.

[21] D. Ruano-Ordás, “Machine Learning-Based Feature Extraction and Selection,” *Applied Sciences*, vol. 14, no. 15, p. 6567, Jul. 2024, https://doi.org/10.3390/app14156567.

[22] L. Berriche and A. Loulizi, “Prediction of failures in the project management knowledge areas using optimized ensemble models in software companies,” *Discover Applied Sciences*, vol. 7, no. 7, p. 718, Jul. 2025, https://doi.org/10.1007/s42452-025-07337-y.

[23] P. Thiengburanathum and P. Charoenkwan, “SETAR: Stacking Ensemble Learning for Thai Sentiment Analysis Using RoBERTa and Hybrid Feature Representation,” *IEEE Access*, vol. 11, pp. 92 822–92 837, August, 2023, https://doi.org/10.1109/ACCESS.2023.3308951.

[24] A. U. Berliana and A. Bustamam, “Implementation of Stacking Ensemble Learning for Classification of COVID-19 using Image Dataset CT Scan and Lung X-Ray,” in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*. Yogyakarta, Indonesia: IEEE, Nov. 2020, pp. 148–152, https://doi.org/10.1109/ICOIACT50329.2020.9332112.

[25] N. Lay-Raby, H. De La Fuente-Mella, and O. Lameles-Corvalán, “Multinomial Logistic Regression to Estimate and Predict the Job Opportunities for People with Disabilities in Chile,” *Information*, vol. 12, no. 9, p. 356, Aug. 2021, https://doi.org/10.3390/info12090356.

[26] O. Oktafia and R. S. A. Nugroho, "Comparison of Support Vector Machine (SVM), XGBoost and Random Forest for Sentiment Analysis of Bumble App User Comments," *Proxies : Jurnal Informatika*, vol. 6, no. 1, pp. 32–46, 2022, https://doi.org/10.24167/proxies.v6i1.12453.

[27] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," *Expert Systems with Applications*, vol. 237, p. 121549, Mar. 2024, https://doi.org/10.1016/j.eswa.2023.121549.

[28] N. Lin, J. Li, and S. Jiang, "A simple but effective method for Indonesian automatic text summarisation," *Connection Science*, vol. 34, no. 1, pp. 29–43, Dec. 2022, https://doi.org/10.1080/09540091.2021.1937942.

[29] A. Almomany, W. R. Ayyad, and A. Jarrah, "Optimized implementation of an improved KNN classification algorithm using Intel FPGA platform: Covid-19 case study," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3815–3827, Jun. 2022, https://doi.org/10.1016/j.jksuci.2022.04.006.

[30] F. E. Arévalo-Cordovilla and M. Peña, "Evaluating ensemble models for fair and interpretable prediction in higher education using multimodal data," *Scientific Reports*, vol. 15, no. 1, p. 29420, Aug. 2025, https://doi.org/10.1038/s41598-025-15388-9.

[31] H. Bihri, L. A. Charaf, S. Azzouzi, and M. E. H. Charaf, "A Robust Stacking-Based Ensemble Model for Predicting Cardiovascular Diseases," *AI*, vol. 6, no. 7, p. 160, Jul. 2025, https://doi.org/10.3390/ai6070160.