IT Journal Research and Development (ITJRD)
Vol.10, No.2, March 2026, E-ISSN : 2528-4053 | P-ISSN : 2528-4061
DOI : 10.25299/itjrd.2025.25550

Job Classification Based on Skills and Qualifications Using Natural Language Processing and Ensemble Learning Methods

Hafiza Oktasia Nasution[1], Dian Ramadhani[2], Mida Aprilina Tarigan[3], Prima Andreas[4], Dewita Suryati Ningsih[5], Arwinence Pramadewi[6]
Department of Business Economics, University of Riau
hafiza@lecturer.unri.ac.id[1], dianramadhani@lecturer.unri.ac.id[2], mida.aprilina@lecturer.unri.ac.id[3], prima.andreas@lecturer.unri.ac.id[4], dewita.suryati@lecturer.unri.ac.id[5], arwinence.pramadewi@lecturer.unri.ac.id[6]

Article Info

Article history:

Received Nov 12, 2025
Revised Dec 26, 2025
Accepted Mar12, 2026

Keyword:

Job Classification
Natural Language Processing
XGBoost
Ensemble Learning
Workforce Analytics

ABSTRACT

This study proposes a job classification framework using Natural Language Processing (NLP) and Ensemble Learning to classify job roles based on required skills and qualifications. A large-scale open-source dataset containing 1.048.576 job postings was utilized, with attributes such as job title, qualifications, skills, company profile, and role. Only relevant attributes were used: skills and qualifications as input features, and role as the target label. Data were filtered to focus on three major job roles—Management, IT, and Digital—resulting in 489,651 relevant entries. Skills were extracted and standardized using GROK AI before feature transformation with MultiLabelBinarizer for one-hot encoding. The XGBoost algorithm was applied for classification under multiple data split configurations (70:15:15, 80:10:10, 70:30, 80:20, 90:10) with random state is 42 and multi-class log loss evaluation. Results showed that the 90:10 configuration achieved the highest accuracy (74.18%), followed by 80:20 with 68.44%. This research demonstrates that ensemble learning effectively handles high-dimensional categorical job data and provides a foundation for automated job classification systems and labor market analysis.

Corresponding Author:

Dian Ramadhani
Departement of Informatics Engineering
University of Riau
HR Soebrantas Street KM 12.5 Simpang Baru Binawidya, Pekanbaru, Indonesia
Email: dianramadhani@lecturer.unri.ac.id

1. INTRODUCTION

In recent years, the digital and information technology sector has experienced rapid growth, along with increasing dependence on technology in various industrial sectors[1]. This requires companies to have a skilled workforce capable of managing rapid changes in the digital world [2]. Therefore, it is important to have an effective system for matching the skills and qualifications of job seekers with available positions in the labor market, especially in the digital field. However, a major challenge currently faced is how to accurately classify digital jobs based on the skills and qualifications required.

The process of matching the right job with the right skills is often a major problem in recruitment. Many systems still rely on criteria that are too general or do not take into account the specific skills required for digital jobs, resulting in a mismatch between available positions and

potential employees. In recent years, approaches based on Natural Language Processing (NLP) and machine learning have been widely applied to tasks such as skill extraction from job descriptions, job category classification, and skill-job mapping. The study Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework uses an extreme multi-label classification (XMLC) framework based on a BERT-based model to extract skills from job descriptions and shows a significant improvement in recall and nDCG [3]. Similarly, NLP and Text Mining for Enriching IT Professional Skills Frameworks utilizes the ESCO taxonomy framework to perform ICT skill extraction and mapping from job postings, positioning skills and qualifications as central components in job classification and matching [4]. Furthermore, the recent study Deep Learning-based Computational Job Market Analysis: A Survey on Skill Extraction and Classification from Job Postings provides a systematic review of NLP methodologies in job market analysis, particularly skill extraction and job posting classification, and emphasizes the need for the development of big data-based job classification models with skill and qualification attributes [5].

To resolve this issue, this study proposes a more comprehensive Natural Language Processing (NLP)-based approach to classify digital jobs, particularly in the fields of management, IT, and digital, by utilizing automatic skill extraction and normalization of relevant qualifications. This approach involves several important stages, beginning with the collection of data from various sources that include information related to qualifications, skills, and job descriptions. Furthermore, filtering is performed on jobs relevant to the main field of research, and skills are extracted by converting the skills in the text into individual skills using a Grok AI-based model, which are then matched with a standardized list of skills using Grok AI. After the preprocessing stage, feature engineering is performed using the one-hot encoding technique on skills and qualifications to prepare the classification data.

This classification system was then trained using the ensemble learning technique XGBoost to classify jobs based on relevant skills and qualifications. By comparing several holdout and cross-validation techniques in the splitting stage and evaluating the model with evaluation metrics such as accuracy, precision, recall, and F1-score, this study aims to produce a more accurate model for classifying digital jobs.

The main innovation in this research is the use of the GROK AI model to extract more in-depth skills automatically with NLP to improve classification accuracy. This approach not only focuses on matching jobs and skills, but also introduces a simpler and clearer qualification normalization process. Thus, this research can make a significant contribution to the development of a more efficient AI-based recruitment system that can be adapted to various industrial sectors in the future.

2. RESEARCH METHOD

The research methodology used is shown in Figure 1. The process begins with data collection, where relevant datasets are gathered for analysis. The collected data then undergoes a pre-processing phase consisting of several steps: feature selection, data filtering, skill extraction, and skill classification. This stage ensures that only relevant and high-quality information is retained for subsequent processing. After pre-processing, the data proceeds to the feature engineering phase, where the extracted features are refined and transformed into formats suitable for model input. The refined dataset is then utilized in the training model phase, where the algorithm learns patterns and relationships between features and target variables. Finally, the trained model undergoes evaluation to assess its performance and accuracy. The results from this evaluation are used to validate the model's effectiveness in predicting or classifying skills based on the collected data.

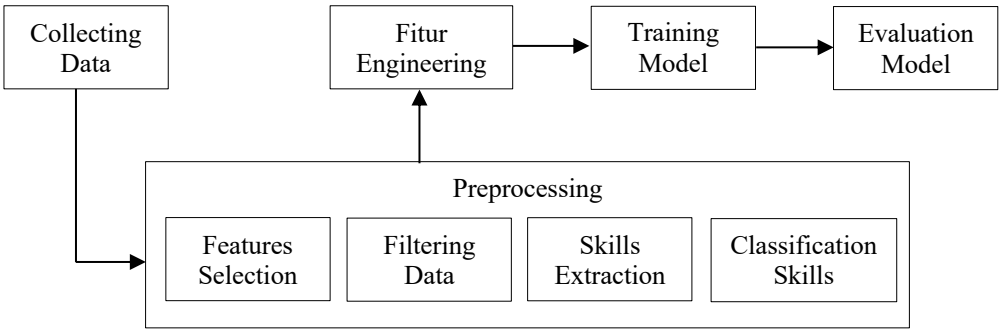


Figure 1. Research Methodology

2.1. Data Collection

In this study, data was collected from open repositories containing information about various professional roles and job characteristics. The dataset includes various attributes, such as job ID, work experience, qualifications, salary range, location, country, geographic coordinates (latitude and longitude), job type, company size, job posting date, preferences, personnel contacts, position, role, job portal, job description, benefits, skills, responsibilities, company name, and company profile. In total, there are 1.048.576 data entries, representing job vacancies from various countries around the world. In this study, only a few attributes relevant to the study objectives were used. The qualifications and skills attributes served as input parameters, while the role attribute was used as an output parameter. Furthermore, a data filtering process was carried out, in which only entries related to the main roles of the study—management, IT, and digital—were retained. After this process, the remaining dataset contained 489.651 entries, ensuring that the data used was truly aligned with the focus of the study. The curated dataset was then further processed through the preprocessing stage to model evaluation. No synthetic data was created in order to maintain the integrity of the original distribution, so that the model evaluation results accurately reflected real-world job classification patterns.

2.2. Preprocessing

At this stage, several preprocessing procedures are applied to prepare the dataset for the classification model [6][7]. The process begins with feature selection, where relevant attributes are selected for the classification task. In this study, Role is used as the output (label), while Qualification and Skills serve as input variables used for prediction.

Furthermore, a data filtering stage is performed to refine the dataset and retain only job roles relevant to the fields of IT and Digital Management. Initially, the available roles are grouped into broader categories with the help of the GROK AI model, facilitating the identification of roles during the filtering process [8]. Then, manual filtering is performed using Microsoft Excel to ensure that the selected roles are appropriate for the scope and context of the study. The skill extraction process converts the original text-based skill descriptions into individual skill entities. Each extracted skill is then matched with the corresponding text-based skill description and stored in a new column named Extract Skills. This step supported by GROK AI, ensures that all relevant skills are accurately identified and systematically recorded in the dataset.

Finally, one-hot encoding is performed using the MultiLabelBinarizer library to convert the extracted Qualifications and Skills parameters into binary numerical representations. This transformation enables machine learning algorithms to process categorical data effectively and improves the accuracy of job role classification based on the identified qualifications and skills.

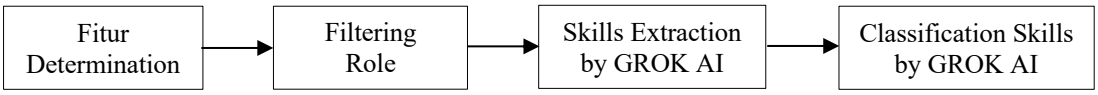


Figure 2. Preprocessing

At this stage, a series of preprocessing steps are performed to prepare the dataset before it is used in the classification model. The first step involves feature selection, which is the selection of relevant attributes to support the classification process. In this study, Role is set as the output variable (label), while Qualification and Skills are used as input variables for prediction purposes.

Furthermore, manual filtering is performed using Microsoft Excel to ensure that the selected roles are appropriate to the context and scope of the study, which focuses on the fields of management, IT, and digital. After this stage, the skills extraction process is performed to convert text-based skill descriptions into individual skill entities. Each extracted skill was then matched with its original skill description and stored in a new column named Extract Skills. The grouping process between the extracted skills and the original text skills was carried out using GROK AI, with the aim of ensuring that all relevant skills could be accurately identified and systematically documented in the dataset.

Table 1. Skills Extraction

Role	Skills	Extract Skills
Accounting Manager	Accounting principles, Financial reporting, Team management, Budgeting, Financial analysis	["Accounting", "Accounting principles", "Budgeting", "Financial analysis", "Financial reporting", "Reporting", "Team management"]

2.3. Feature Engineering

In this study, the feature engineering stage was designed to convert categorical variables into machine-readable numeric formats, ensuring that the model could effectively interpret and process complex multi-skill data. To achieve this, the One-Hot Encoding method was implemented using the MultiLabelBinarizer library. This approach was selected due to its robustness in representing categorical attributes with multiple labels, particularly within datasets containing overlapping qualifications and diverse skill sets [9][10].

Each categorical value, such as Python or Project Management, was converted into an individual binary feature column, where the presence of a specific qualification or skill was represented by “1” and its absence by “0.” This binary encoding mechanism allows the model to independently recognize each feature without introducing artificial ordinal relationships among categorical categories, thereby preserving the semantic integrity of the original data.

The resulting encoded matrix forms a structured high-dimensional representation that preserves the uniqueness of each qualification and skill attribute. This representation enables the classification model to identify correlations, co-occurrence patterns, and hidden relationships among multiple competencies with greater precision while maintaining consistency during training and evaluation.

Although One-Hot Encoding is widely accepted as a standard and effective method for categorical feature transformation, its implementation may introduce sparsity issues when applied to datasets containing thousands of unique skill variations. In such cases, the generated feature space becomes highly sparse because most binary columns contain zero values for the majority of records. This sparsity can increase memory consumption, computational cost, and processing time, particularly when handling large-scale organizational datasets with extensive skill vocabularies. Moreover, sparse representations may reduce learning efficiency for certain machine learning algorithms, especially those sensitive to feature dimensionality.

To address this challenge, several preprocessing controls were applied before encoding. First, skill normalization was performed to merge semantically equivalent terms and eliminate inconsistent naming conventions, such as abbreviations, synonyms, and duplicate expressions referring to the same competency. Second, low-frequency skill attributes that appeared only in a very limited number of records were filtered to reduce feature redundancy and avoid unnecessary dimensional expansion. This filtering process ensured that only representative and analytically meaningful skills were retained in the final feature space.

In addition, the use of MultiLabelBinarizer remains suitable for this study because the objective emphasizes transparent and interpretable feature representation, where each skill must remain explicitly identifiable for mismatch analysis. Unlike embedding-based approaches that compress semantic information into dense vectors, binary encoding preserves direct traceability of individual skill contributions, which is essential for explaining classification outcomes in organizational decision-making contexts

As a result, the final encoded matrix provides a balance between interpretability, scalability, and computational feasibility. This allows the classification model to learn multi-attribute patterns effectively while minimizing the negative impact of sparsity in processing large-scale job-related datasets.

Figure 3. Feature Engineering

2.4. Extreme Gradient Boosting (XGBoost)

XGBoost is a decision tree-based machine learning algorithm that uses a gradient boosting framework to generate powerful predictive models [11][12]. This algorithm uses a second-order Taylor expansion of the loss function and adds regularization terms, such as the gamma (γ) and lambda (λ) parameters, to control tree complexity and prevent overfitting. As a result, XGBoost not only focuses on improving prediction performance, but also maintains model stability and generalization capabilities.

One of XGBoost's main strengths is its ability to efficiently handle large datasets, multicategorical features, and potentially imbalanced classes[13][14]. This makes it the right choice for classifying skill- and qualification-based jobs in this study, as the dataset involves many features such as skill extraction results, qualifications, and roles, and requires modeling that can effectively distinguish between roles.

The objective function of the Extreme Gradient Boosting algorithm is written as follows:

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \tag{1}$$

Where $L(y_i,F(x_i))$ is the loss function that measures how far the model prediction $F(x_i)$ is from the actual value y_i. Meanwhile, $\Omega(h_m)$ is the regularization term used to control model complexity. Regularization is defined as follows

$$\Omega(h) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{2}$$

The γ parameter controls the number of leaf T in each decision tree, while λ adjusts the penalty for the weight w at each leaf. By adding these components, XGBoost not only minimizes prediction errors, but also keeps the model from becoming too complex, thereby preventing

overfitting. This objective function allows XGBoost to balance accuracy and generalization. Each iteration of the algorithm attempts to improve on the previous model's error by adding new trees optimized based on the gradient of the loss function, while maintaining stability through structural regularization. This equation forms the mathematical basis that makes XGBoost more efficient and stable than conventional Gradient Boosting Machine (GBM) algorithms, and is the reason why this algorithm is widely used in various classification studies, including in the classification of job roles based on qualifications and skills.

Several studies have demonstrated the success of using XGBoost in similar domains. For example, in the study "Predicting Skill Shortages in Labor Markets: A Machine Learning Approach," researchers used XGBoost to predict skill shortages in the Australian labor market and successfully obtained a macro-F1 score of around 83% [15]. Furthermore, Application of the XGBoost Model with Hyperparameter Tuning for Industry Classification for Job Applicants shows the application of XGBoost on a dataset of job applicants from Kaggle and produces an accuracy of around 0.89 to 0.90 after hyperparameter tuning [16]. Then, Job Recruitment Analysis based on XGBoost Decision Tree uses XGBoost for job position analysis and recruitment requirements, confirming that the algorithm can assist in recruitment and job matching for graduates [17].

In the context of this research, XGBoost is a strategic choice for several reasons. First, the numerous input features based on the extraction of individual skills and qualifications require an algorithm capable of handling many variables and complex interactions between features. Second, this model effectively controls tree complexity, making it more resistant to overfitting, which is important when the dataset has many one-hot encoded result columns. Third, XGBoost's ability to perform hyperparameter tuning, such as n_estimators, learning_rate, max_depth, and gamma, allows for optimal adjustment according to the characteristics of the research dataset.

2.5. Model Evaluation

The Confusion Matrix is one of the evaluation methods used to measure the performance of a classification model by comparing the model's prediction results with the actual values of the test data [18][19]. This matrix is represented in the form of a two-dimensional table consisting of four main elements, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), as shown in Table 2.

Table 2. Model Evaluation

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

These values are then used to calculate key performance metrics that serve as indicators of classification model accuracy, namely Accuracy, Precision, Recall, and F1-Score. Furthermore, the elements in the confusion matrix are used to determine these important performance metrics in accordance with the mathematical definitions described in Equations 3–6.

$$accuracy = \frac{T_p+T_n}{T_p+T_n+F_p+F_n} \tag{3}$$

$$precision = \frac{T_p}{T_p+F_p} \tag{4}$$

$$recall = \frac{T_p}{T_p+F_n} \tag{5}$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \quad (6)$$

The evaluation framework includes a number of interrelated metrics for assessing model performance. Accuracy describes the proportion of correct predictions relative to all test data, calculated based on the diagonal elements of the confusion matrix. Precision shows the ratio between the number of samples that were correctly classified and the total number of samples predicted in each class. Meanwhile, recall or sensitivity measures the model's ability to recognize all actual samples that truly belong to a class, by comparing the number of correct predictions to the total actual samples in that class.

A high precision value indicates that the model has a low false positive error rate, while a high recall value indicates the model's ability to detect most of the relevant samples in each class. To balance these two metrics, the F1-score is used, which is the harmonic mean between precision and recall that provides a more comprehensive picture of the classification model's performance.

3. RESULTS AND ANALYSIS

This stage describes the training process design, testing, and analysis strategy used to classify job roles based on qualifications and skills using the Extreme Gradient Boosting (XGBoost) algorithm. This algorithm was chosen for its ability to efficiently handle large datasets with many categories, as well as its ability to minimize classification errors through a gradient boosting mechanism accompanied by model complexity regulation[11][20].

The experimental framework adopts a holdout strategy with various data distribution ratios in both train–validation–test and train–test settings to assess the influence of data proportion differences on predictive performance. The selected schemes comprise 70:15:15 and 80:10:10 with validation support, together with 70:30, 80:20, and 90:10 for direct training-to-testing evaluation without separate validation data.

Model evaluation was performed using the multi-class logarithmic loss (mlogloss) metric and random state 42. The mlogloss metric was used to assess the quality of prediction probabilities in multi-class classification and is sensitive to the calibration level of the model [Farhan]. Meanwhile, random state 42 was applied to ensure reproducibility, following common practice in scientific literature that emphasizes the importance of controlling random elements so that the data partitioning process and evaluation results can be replicated consistently [Shin].

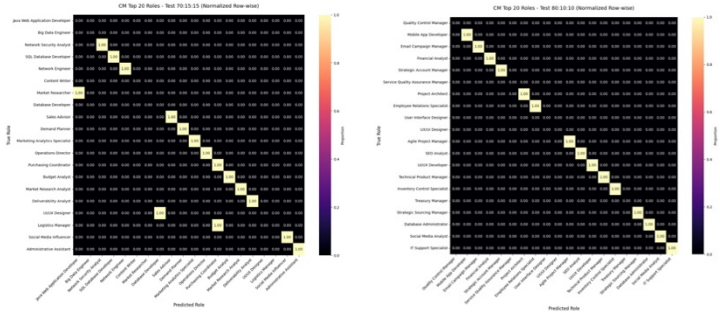
Tabel 2. Model Performance of Holdout Evaluation

Model	Ratio	Accuracy	Precission	Recall	F1-Score
XGBoost	70:15:15	66.12%	65.37%	65.57%	65.44%
	80:10:10	68.44%	68.24%	68.44%	68.31%
	70:30	65.98%	65.68%	65.98%	65.79%
	80:20	68.44%	67.83%	68.44%	68.06%
	90:10	74.18%	74.18%	74.18%	74.18%

As shown in the table, the XGBoost model achieved its highest performance under the 90:10 training–testing ratio, with accuracy, precision, recall, and F1-score all reaching 74.18%. This result indicates that when the model is trained with a larger proportion of data, it benefits from more learning instances, leading to enhanced predictive performance across all metrics. The balanced performance across precision, recall, and F1-score further emphasizes the model's ability to generalize well on unseen data when trained on more data. Instead, the lowest performance was recorded in the 70:30 configuration, with accuracy at 65.98%, precision at 65.68%, recall at 65.98%,

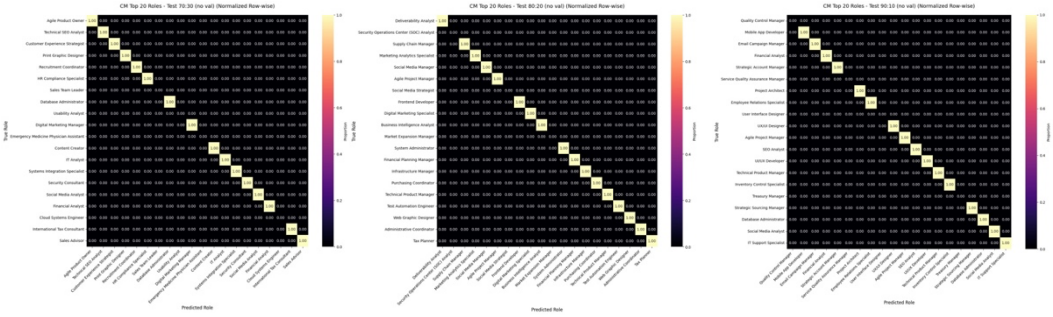
and F1-score at 65.79%. These values suggest that reducing the proportion of training data limits the model's capacity to capture complex feature patterns and learn effectively. The 70:15:15 and 80:20 configurations show moderate improvements in all metrics, with accuracy increasing from 66.12% to 68.44%, and precision improving from 65.37% to 67.83%. These results indicate that adding validation data helps prevent overfitting and stabilizes learning outcomes. The 80:10:10 configuration yields the same accuracy value of 68.44%, but with a slight drop in precision (from 68.24% to 67.83%) while recall improves to 68.44%. This suggests that adding more validation data does not significantly alter the model's generalization behavior beyond a certain point, particularly for the precision-recall balance.

In summary, the findings highlight that the XGBoost algorithm is sensitive to the distribution of training and testing data. As the training proportion increases, the model's performance improves, especially in the 90:10 ratio where all metrics reach their peak values. However, the improvement becomes more pronounced only after the 80:20 ratio, reflecting a trade-off between model complexity and data sufficiency. The precision, recall, and F1-score also follow similar trends, showing a clear benefit when the model is trained on larger datasets.



(a) (b)

Figure 3. The 20 Most Frequent Classes Train–Validation–Test



(a) (b) (c)

Figure 4. The 20 Most Frequent Classes Train–Validation–Test

The confusion matrix visualizations presented in Figures 3 and 4 provide a more detailed understanding of class-level prediction behavior across different data split configurations. Overall, most dominant classes are concentrated along the diagonal, indicating that the XGBoost model correctly classified a substantial proportion of the top 20 job roles under all experimental settings. This confirms that the model successfully learned discriminative feature patterns from the encoded skill representations.

However, a closer examination of the confusion matrices reveals that several classification errors occur among job roles with highly overlapping skill requirements. For example, technical roles such as UI/UX Designer, exhibit potential classification proximity because they share common

competencies including interface development, design tools, prototyping, and application-oriented workflows. Although these roles belong to different professional categories, the encoded binary skill representation captures many identical attributes, which reduces feature separability and increases the likelihood of prediction overlap.

4. CONCLUSION

This research successfully demonstrates that by utilizing a comprehensive Natural Language Processing (NLP) approach for skill extraction and qualification normalization, digital job classification can be significantly improved. The results show that the XGBoost model, when trained with an adequate proportion of data, can achieve high performance in classifying digital jobs based on the skills and qualifications required. As discussed in the introduction section, this study aimed to address the challenge of matching digital jobs with the right candidates. The findings in the results section confirm that the proposed system is capable of achieving this goal effectively, making a substantial contribution to the development of AI-based recruitment systems.

Looking forward, this research opens several prospects for further development. Future work could focus on enhancing the feature extraction process, integrating additional data sources such as company-specific skill requirements, or refining the model with more advanced NLP techniques like transformers and large language models (LLMs). Additionally, the system could be further tested across different industries to explore its adaptability and scalability. Ultimately, this study paves the way for the next generation of recruitment systems that can more accurately match job seekers with digital positions based on their qualifications and skills.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to the University of Riau through its research and community service institution for the support provided in this research under the Research Scheme for the Capacity Building of Young Lecturers (RIPEKDOM) with number 29167/UN19.5.1.3/AL.04/2025.

REFERENCES

[1] V. S. Litvinenko, "Digital economy as a factor in the technological development of the mineral sector," *Nat. Resour. Res.*, vol. 29, no. 3, pp. 1521–1541, 2020.

[2] M. J. Sousa and D. Wilks, "Sustainable skills for the world of work in the digital age," *Syst. Res. Behav. Sci.*, vol. 35, no. 4, pp. 399–405, 2018.

[3] A. Bhola, K. Halder, A. Prasad, and M. Y. Kan, "Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework," *COLING 2020 - 28th Int. Conf. Comput. Linguist. Proc. Conf.*, pp. 5832–5842, 2020, doi: 10.18653/v1/2020.coling-main.513.

[4] D. Zare, L. Fernandez-Sanz, V. Pospelova, and I. López-Baldominos, "NLP and Text Mining for Enriching IT Professional Skills Frameworks," *Appl. Sci.*, vol. 15, no. 17, pp. 1–20, 2025, doi: 10.3390/app15179634.

[5] E. Senger, M. Zhang, R. van der Goot, and B. Plank, "Deep Learning-based Computational Job Market Analysis: A Survey on Skill Extraction and Classification from Job Postings," *NLP4HR 2024 - 1st Work. Nat. Lang. Process. Hum. Resour. Proc. Work.*, no. Nlp4hr, pp. 1–15, 2024.

[6] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intell. data Anal.*, vol. 1, no. 1–4, pp. 3–23, 1997.

[7] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*, vol. 72. Springer, 2015.

[8] M. E. de Carvalho Souza and L. Weigang, "Grok, gemini, chatgpt and deepseek: Comparison and applications in conversational artificial intelligence," *Intel. Artif.*, vol. 2, no. 1, 2025.

[9] M. M. Abushaega, O. Y. Moshebah, A. Hamzi, and S. Y. Alghamdi, "Multi-objective sustainability optimization in modern supply chain networks: A hybrid approach with

federated learning and graph neural networks," *Alexandria Eng. J.*, vol. 115, pp. 585–602, 2025.

[10] W. Gao, Z. Ding, J. Lu, and Y. Wan, "Low-carbon information quality dimensions and random forest algorithm evaluation model in digital marketing," *Sci. Rep.*, vol. 14, no. 1, p. 22416, 2024.

[11] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R Packag. version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[12] T. Chen, T. He, M. Benesty, and V. Khotilovich, "Package 'xgboost,'" *R version*, vol. 90, no. 1–66, p. 40, 2019.

[13] J. Dong, Y. Chen, B. Yao, X. Zhang, and N. Zeng, "A neural network boosting regression model based on XGBoost," *Appl. Soft Comput.*, vol. 125, p. 109067, 2022.

[14] X. Zhang, C. Yan, C. Gao, B. A. Malin, and Y. Chen, "Predicting missing values in medical data via XGBoost regression," *J. Healthc. informatics Res.*, vol. 4, no. 4, pp. 383–394, 2020.

[15] N. Dawson, M.-A. Rizoiu, B. Johnston, and M.-A. Williams, "Predicting skill shortages in labor markets: A machine learning approach," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 3052–3061.

[16] A. A. Syahputra and R. E. Saputro, "Application of the XGBoost Model with Hyperparameter Tuning for Industry Classification for Job Applicants," *Sinkron*, vol. 8, no. 3, pp. 1920–1931, 2024, doi: 10.33395/sinkron.v8i3.13840.

[17] J. Chen, Y. Chen, Y. Liao, J. Mu, G. Li, and J. Li, "Job Recruitment Analysis based on Xgboost Decision Tree," *Int. J. Soc. Sci. Public Adm.*, vol. 2, no. 3, pp. 392–396, 2024, doi: 10.62051/ijsspa.v2n3.56.

[18] M. A. Khan, M. O. Khan, H. Noureen, M. S. Khan, and M. Fawad, "A Hybrid Transformer and CNN-Based Approach for Classifying Mental Health Disorders from Social Media Data," 2025.

[19] M. Tajrian, A. Rahman, M. A. Kabir, and M. R. Islam, "Analysis of child development facts and myths using text mining techniques and classification models," *Heliyon*, vol. 10, no. 17, 2024.

[20] A. I. A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, and A. El-Shafie, "Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia," *Ain Shams Eng. J.*, vol. 12, no. 2, pp. 1545–1556, 2021.