Prediction of Student Job Readiness Using MLP and XGBoost Method

1st Herman Gemilang
School of Computing
Telkom University
Bandung, Indonesia
hermangemilang@student.telkomuniversity.ac.id

2nd Kemas Muslim Lhaksmana
School of Computing
Telkom University
Bandung, Indonesia
kemasmuslim@telkomuniversity.ac.id

***Abstract*—Student job readiness is a crucial issue that requires serious attention in the university environment. This problem is crucial because universities need to identify existing students whose job readiness is low to prepare them to enter the workforce. The manual detection process has significant limitations, such as the need for psychologists, which requires substantial time, effort, and money. Therefore, implementing machine learning can address these challenges more effectively. This study aims to analyze and compare the performance of two algorithms, Multi-Layer Perceptron (MLP) and Extreme Gradient Boosting (XGBoost), in predicting student job readiness using a dataset of XYZ University graduates, which includes student data and post-graduation job readiness surveys. Additionally, this study seeks to evaluate the effectiveness of these models compared to previous research findings, specifically those obtained using the Multinomial Logistic Regression model. The results showed that the XGBoost model consistently performed better than the MLP model at various data-sharing ratios, with XGBoost achieving an accuracy of 89% compared to MLP 82%. Furthermore, XGBoost precision, recall, and F1-Score are higher than those of MLP, indicating XGBoost superior ability to generalize models. Enhancing performance is achieved using SMOTEENN technique, which aids in balancing the data and boosting the model's effectiveness. The comparison with previous research, where the best Multinomial Logistic Regression model achieved an accuracy of 53.9%, highlights the significant improvement in accuracy with XGBoost and underscores the effectiveness of modern machine learning techniques in classifying students' job readiness based on when they got a job.**

Keywords—multi-layer perceptron (MLP), extreme gradient boosting (XGBoost), hyperparameter, job readiness, smoteenn

I. Introduction

Student job readiness is a crucial issue in the university environment, considering the importance of preparing graduates that are ready to compete in the job market. Many students who have completed their academic education are still not ready to enter the workforce [1]. Factors such as the incompatibility of competencies provided by educational institutions with those needed by the company, lack of work experience, and limitations in the development of soft skills are the main causes of this problem [2]. Universities face the challenge of detecting students who are not ready to work manually because this process requires significant time, effort, and cost. Therefore, the application of machine learning is the right solution to overcome this limitation and help universities detect job readiness more efficiently and accurately.

Machine learning, as a subfield of artificial intelligence, allows computers to process data and learn from it without the need for explicit programing [3]. The potential of machine learning to provide objective and in-depth analysis of huge amounts of data and its ability to process information quickly make it relevant for solving problems in various aspects of life. In this study, two types of machine learning algorithms were used: Multi-Layer Perceptron (MLP) and Extreme Gradient Boosting (XGBoost). MLP is a type of artificial neural network that is widely used for classification and prediction problems [4]. The XGBoost algorithm is one of the most powerful and efficient machine learning algorithms for classification and prediction problems [5]. The selection of these two algorithms is based on the similarity of each algorithm, which has advantages in handling classification and prediction problems.

This study aims to analyze and compare the performance of MLP and XGBoost in predicting student job readiness using a dataset of XYZ University graduates, which includes student data and post-graduation job readiness surveys. Additionally, this study seeks to evaluate the effectiveness of these models compared to previous research findings, specifically those obtained using the Multinomial Logistic Regression model. As a first step, algorithms are trained using graduates' historical data to understand patterns and trends in student job readiness. After the training process, algorithms are applied to the new data to predict the waiting period of graduates before they enter the workforce. In other words, the algorithms estimate how long graduates will need to be ready to enter the workforce. Furthermore, it investigates the effectiveness of SMOTE-ENN (Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors) in enhancing model performance, offering insights for adaptive educational strategies. SMOTE-ENN is utilized to address class imbalance and improve the efficacy of the models.

II. Related Work

A study investigating job readiness has previously been conducted using Multinomial Logistic Regression (MLR) [6]. This study analyzed the impact of various factors, including soft and hard skills, on students' preparedness for the workforce. Using MLR, the researchers found significant correlations between these factors and students' job readiness, providing valuable insights for talent management and human resources. This study contributes to the growing body of literature on predictive modeling for workforce preparedness and highlights the importance of using diverse methodologies to address complex research questions in the field of people analytics. The MLR study achieved an accuracy rate of 53.9%, further emphasizing its effectiveness in predicting job readiness.

However, despite the achievement of the MLR model, there is still potential for improvement in predictive accuracy. Some researchers have turned to MLP algorithms for predictive case studies [7], [8], whereas others have explored the use of the XGBoost algorithm [9], [10]. The use of both algorithms showcases their promising capabilities in supporting various prediction case studies.

2024 International Conference on Data Science and Its Applications (ICoDSA) | 979-8-3503-6535-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICODSA62899.2024.10652042

Based on the results of performance evaluation using the MLP algorithm in research [7], it proved effective in predicting software defects by achieving an average accuracy of 87.5% from 12 datasets. The dataset is taken from software data residing in NASA's systems. Based on another study [8], the MLP algorithm also produced a model with an accuracy of 92.3%. The dataset used in this study was student data from three colleges in India. In this study, the authors emphasize the importance of overcoming unbalanced data to obtain better results.

In addition to implementing the MLP algorithm, the author uses another algorithm that is very suitable for prediction case studies, namely the XGBoost algorithm. In research [9], the accuracy of the model using the XGBoost algorithm reached 78.75% on the ASSISTments dataset and consistently became the algorithm with the best accuracy in other datasets. Meanwhile, the accuracy of the XGBoost algorithm in Research [10], which aims to develop a model to map student profiles reaching 88.96%, is in second place as the most effective algorithm under the LightGBM algorithm. As a new model was discovered in 2016, XGBoost should be further developed as it has enormous potential in classification and prediction case studies.

III. System Design

This study aims to find the most effective model for analyzing the job readiness of students at XYZ University. In analyzing this, the MLP and XGBoost algorithms are applied to make predictions. The system design is shown in Fig. 1, which describes the overall development of the designed system.

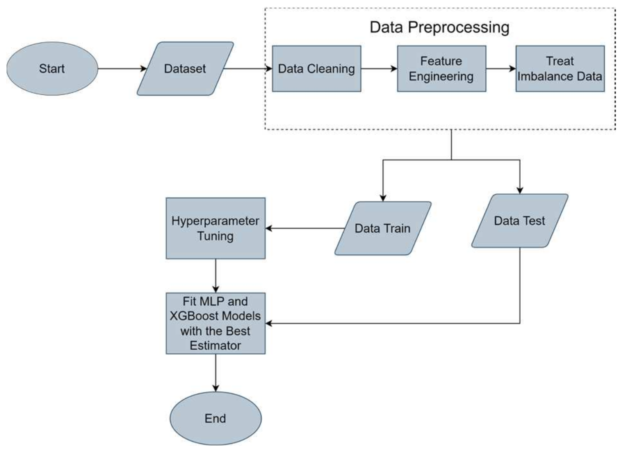

Fig. 1. System Design Flow.

A. Dataset

The dataset used in this study is alumnus data released by XYZ University. These data includes student data and survey questions provided after graduation. The student ID attribute will be a unique number used to identify the other attribute. The dataset consists of 8543 rows of data with 48 attributes loaded in the CSV format. TABLE I provides details on some of the key attributes and their descriptions:

TABLE I. Dataset Attribute

Atrribute Name	Description
NIM	Student ID
Number_of_Responding_Companies	Responding companies
Required_Teamwork_Competence	Teamwork competence
Required_Communication_Competence	Communication competence
Required_Self_Development_Competen ce	Self-development competence
Study_Program	The study program undertaken by the student
Emphasis_on_Demonstration_Learning_ Method	Demonstration learning
Emphasis_on_Discussion_Learning_Met hod	Discussion learning.
Emphasis_on_Fieldwork_Learning_Met hod	Fieldwork learning
Emphasis_on_Participation_in_Research _Project_Method	Research project learning
Target	Job readiness status

B. Data Preprocessing

1) Data Cleaning

The data cleansing phase involves handling missing values, duplications, and outliers. Missing values are identified and deleted. In addition, the associated attribute is removed if the number of missing values is significant. Duplication problems are identified on the basis of the Student Identification Number (NIM), which is a unique number for each student. Rows containing duplicate data are removed, retaining the first row containing the NIM. Outliers were detected by computing the interquartile range (IQR) and visualizing the feature columns with boxplots. Rows with values below the first quartile (Q1) and above the third quartile (Q3) are considered outliers.

2) Feature Engineering

Feature engineering is an important step in the data processing process that improves the performance of machine learning models by creating and selecting relevant attributes. This process begins with the creation of a new attribute based on an existing attribute. These new attributes are created to capture additional information that may not be visible from the original attribute, such as ratios or interactions between two attributes that can reveal more complex relationships in the data.

After creating a new attribute, the next step is to calculate the correlation between attributes. This correlation calculation helps us understand the extent to which one attribute is related to another. The correlation matrix is used to identify attributes that have high correlation, which can lead to multicollinearity in the model. Attributes with a high correlation to other attributes may be considered for removal or merging.

Based on the results of the correlation calculation, attributes that are considered irrelevant or have a very high correlation with other attributes are removed from the dataset. The removal of this attribute reduces the complexity of the model and improve its performance and interpretability. In addition, attributes that do not add value to the model are eliminated, so only the most relevant attributes are used in further analysis.

To handle string attributes, encoder labels are used to convert categorical values into a numeric format. This process is important for machine learning models to process data correctly because most machine learning algorithms only accept numerical input. Each category in a string attribute is converted into a unique numeric value, so the model can use that information more effectively.

Furthermore, the waiting period for jobs was classified into three classes based on the average waiting period for alumni taken from the XYZ University Tracer Study Data 2020-2021, which was 3.46 months. The three classes are students who found employment before graduation, those who found employment less than 3.46 months after graduation, and those who found employment more than 3.46 months after graduation. This classification helps in understanding and modeling the factors that affect the speed at which students get jobs after graduation, and in Evaluating the model's performance in predicting student job waiting periods.

3) Treat Imbalance Data

To address data imbalance in this study, the SMOTEENN method was employed, merging the Synthetic Minority Oversampling Technique (SMOTE) with Edited Nearest Neighbors (ENN). SMOTE increases the number of minority class samples by creating synthetic samples through interpolation between minority samples and their closest neighbors [11]. Then, ENN cleans up the dataset by removing samples misclassified by k nearest neighbors, thereby reducing noise and ensuring cleaner data. The SMOTEENN implementation is performed using the "imblearn" library in Python, resulting in a more balanced and quality dataset, which improves the performance of machine learning models.

C. Hyperparameter Tuning

In this section, hyperparameter tuning is performed to improve the model performance. For the Multi-Layer Perceptron MLP model, tuning is performed on several main hyperparameters, namely hidden layer size, activation function, optimization algorithm, and learning rate. For the XGBoost model, optimization was carried out on hyperparameters such as learning rate and maximum depth of the Tree, minimum weight for child nodes, and L1 regulation and L2 regulation. Hyperparameter tuning finds the combination of parameters that provide the best performance for each model.

D. Model Construction

The research leveraged two powerful machine learning algorithms for classification and prediction, namely Multi-Layer Perceptron (MLP) and XGBoost. The selection of MLP and XGBoost is based on their respective advantages in handling various classification and prediction problems, as well as their ability to process data with diverse characteristics so that they are suitable for implementation in the case of job readiness prediction.

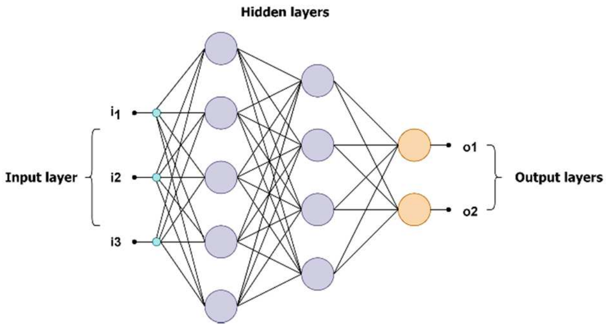

Fig. 2. MLP Architecture.

1) Multi-layer Perceptron (MLP)

Multilayer perceptron (MLP) is a type of neural network architecture consisting of several layers that are connected [12]. MLP identifies complex data patterns by passing the input through multiple hidden layers and generating an output at the final layer. With its advantages in overcoming complex data structures, MLP algorithms have become effective solutions in areas such as classification, prediction, and natural language processing.

As illustrated in Fig. 2, MLP consists of an input layer, a hidden layer, and an output layer. In the hidden layer, each neuron is connected to each neuron in the previous layer and after it. Neurons also have weights and biases that are regulated during the training process. The result of the summation of each neuron will subsequently be fed to the activation function. This function determines whether the neuron must be "active" or not. The activation function can be seen in the following (1):

$$a_j = f(z_j) \tag{1}$$

where a_j is the output of the neuron j after the activation function $f(z_j)$ is applied. The activation function used after going through the hyperparameter tuning process is tanh. This process, called feedforward, continues until it finally reaches the output layer. The output layer produces predictions or output values. Here is (2) for the output layer:

$$y_k = f(\sum_{j=1}^{m} a_j w_{jk} + b_k) \tag{2}$$

The output a_j of neurons j in the hidden layer is used as input. The input is subsequently multiplied by the weight w_{jk} linking neuron j in the hidden layer to neuron k in the output layer and is further adjusted by the bias (b_k). The result of this summation is then applied to the activation function f to produce the output y_k. Output y_k is a class prediction for the working readiness of the model after the activation process on the total number of inputs, weights, and biases. Fig. 2 explains the architecture of the MLP algorithm.

2) Extreme Gradient Boosting (XGBoost)

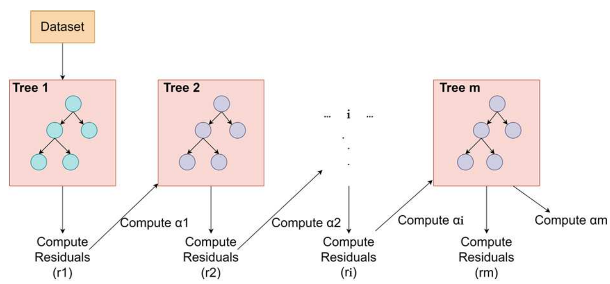

Fig. 3. XGBoost Architecture.

Extreme Gradient Boosting (XGBoost) is a popular machine learning algorithm for solving regression and classification problems [13]. This algorithm is a development of the Gradient Boosting Decision Tree (GBDT) algorithm, which has better performance in terms of accuracy and computational time [14]. XGBoost works by controlling the complexity of the model and pruning at a later stage to avoid overfitting. Another advantage of the XGBoost algorithm lies in its careful optimization and scalability, so it can cope with increased data volume or computational complexity without significantly sacrificing performance or computational time. This algorithm also has awareness of missing values and 0 values in the dataset by applying effective feature engineering techniques so that this model can have advantages that other algorithms do not have.

As shown in Fig. 3, XGBoost is a machine learning algorithm based on ensemble learning techniques that employ a boosting approach. This method involves building a series of weak models, such as simple decision trees. Each decision tree is built in sequence, where each new model addresses the prediction errors of the preceding model by calculating the residuals between the actual values and the model's predictions. In addition, a learning rate is applied to each new model to control its contribution in correcting the errors of the previous models.

In this method, an objective function is needed to evaluate how well the resulting model matches the training data. The main characteristic of this objective function consists of two parts, namely, the training error value and the regularization value, as shown in the following (3):

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3}$$

Specifically, l is a loss function that measures the difference between the prediction $\hat{y}_i$ and the actual value y_i. Additionally, θ is the model parameter, n is the amount of data, K and is the total number of decision trees in the ensemble model. While $\Omega(f_k)$ is a regularization function that measures the complexity of the model to prevent overfitting. The Ω regularization function in XGBoost can be shown in (4) below:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{4}$$

where, T is the number of leaves on the decision tree, which indicates the complexity of the tree structure. γT is a parameter that helps manage the number of leaves in a decision tree, facilitating pruning to avoid overfitting by structuring the tree. λ is a parameter that regulates the L2 regularization strength applied to the leaf weights (w_j). By increasing the value λ, the regulation of L2 becomes stronger, which reduces the magnitude of the weight and prevents overfitting. The higher the value λ, the stronger the L2 regulation, and the simpler the model.

TABLE II. CONFUSION MATRIX

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

E. Evaluation

To assess the quality of the developed model, it will be evaluated using the K-Fold Cross-Validation method, performed over five iterations. In each iteration, four subsets are used to train the model, and one subset is used to test the model. Model evaluation will include metrics such as accuracy, precision, recall, and F1-score, each of which provides a different view of the performance of the student job readiness prediction system. The process of calculating this evaluation metric is inseparable from the confusion matrix, which is used to evaluate the performance of prediction models. This confusion matrix consists of four terms: true positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [15], [16], [17], [18]. Further explanations of the confusion matrix are presented in TABLE II.

Using the confusion matrix in TABLE II, the performance of the classification model can be evaluated by calculating Accuracy, Precision, Recall (or Sensitivity), and F1-Score. The formulas can be seen in the following equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$f1\ Score = \frac{2 \times Precision \times Recall}{Precision \times Recall} \tag{8}$$

IV. RESULT AND DISCUSSION

This study adopts two optimal test scenarios to evaluate XGBoost and MLP models. The first scenario involves testing both models with and without using the SMOTEENN technique to deal with data imbalances, thus allowing a comparison of model performance under different conditions, including those similar to previous MLR-based research. The second scenario focuses on finding the split data that produces the best accuracy in K-Fold Cross Validation. By comparing model performance in these two scenarios, this study aims to

Fig. 4. Target Classes Distribution.

TABLE III. MODEL WITHOUT SMOTEENN

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.48	**0.49**	**0.40**	**0.39**
MLP	0.44	0.39	0.37	0.34
MLR	**0.55**	0.35	0.38	0.38

TABLE IV. MODEL WITH SMOTEENN

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	**0.84**	**0.82**	**0.71**	**0.75**
MLP	0.78	0.75	0.59	0.62
MLR	0.54	0.54	0.53	0.54

provide deeper insights into the effectiveness of various approaches in improving the accuracy of student job readiness predictions. The results showed that certain combinations of models and data balancing techniques can provide more optimal results in analyzing student job readiness.

A. Scenario 1

The target class distribution is one of the main focuses that must be considered to obtain optimal results. Fig. Figure 4 shows that there are significant differences between classes 2 and 0 and 1. In the first scenario, this study tested the XGBoost and MLP models with two different approaches, using and without the SMOTEENN technique. In the first approach, models are trained and tested using original, unbalanced data to determine the baseline performance of each model. In the second approach, SMOTEENN is applied to the training data before the model is trained to measure the impact of data balancing on model performance. As mentioned in the design of the system, SMOTE-ENN is a hybrid method for oversampling and undersampling data. Next, the model is evaluated with accuracy, precision, recall, and F1-score metrics.

As shown in TABLE III, while the MLR model achieved a higher accuracy of 55% compared to XGBoost 48% and MLP with 44%, it is notable that the MLR model had lower precision, recall, and F1-score values than both XGBoost and MLP. Despite its higher accuracy, MLR performance in terms of precision, recall, and F1-score metrics indicate a potential issue with its ability to correctly identify positive cases and avoid false positives and negatives compared with XGBoost and MLP.

TABLE V. EVALUATION COMPARISON ACROSS DATA SPLITS

Model	Split (Train:test)	Accuracy	Precision	Recall	F1-Score
XGBoost	60:40	0.84	0.82	0.70	0.74
	70:30	0.84	0.82	0.71	0.75
	80:20	0.87	0.81	0.77	0.79
	90:10	**0.89**	**0.84**	**0.77**	**0.82**
MLP	60:40	0.77	0.75	0.54	0.58
	70:30	0.77	0.75	0.59	0.62
	80:20	0.78	0.70	0.54	0.56
	90:10	0.85	0.84	0.70	0.75
MLR	60:40	0.52	-	-	-
	70:30	0.54	0.54	0.53	0.54
	80:20	0.52	-	-	-
	90:10	0.33	-	-	-

These discrepancies highlight the importance of considering multiple evaluation metrics when assessing model performance.

Based on TABLE IV, the implementation of SMOTEENN significantly improved the performance of all three models. The XGBoost algorithm achieved an accuracy of 84%, MLP achieved 78%, and MLR achieved 54%. Moreover, XGBoost's precision, recall, and F1-score consistently outperformed those of MLP and MLR. These results were obtained after the same hyperparameter tuning process as in the previous experiment.

B. Scenario 2

In the first scenario, it appears that the accuracy of the models using SMOTE-ENN is better. In the second scenario, a process of sharing data is carried out to find the best model. This process will be carried out in four experiments with data-sharing ratios of 60:40, 70:30, 80:20, and 90:10, where the first number indicates the percentage of training data, and the second number indicates the percentage of test data. Next, the model will be evaluated using accuracy metrics.

Based on the results presented in TABLE V, notable differences in performance were observed between XGBoost, MLP, and MLR across different data splits. XGBoost consistently demonstrated superior accuracy, precision, recall, and F1-score compared with MLP and MLR across all tested scenarios. For instance, XGBoost achieves an accuracy of 84% in the 60:40 data split, whereas MLP achieves only 77%. Meanwhile, MLR, used in previous research, exhibits less consistent performance, with an accuracy of only 54% in the 70:30 data split. These results underscore the robustness and efficacy of XGBoost in classification tasks, indicating its ability to deliver accurate predictions regardless of variations in the distribution of training and test data.

V. CONCLUSION

The comparison of performance between XGBoost, MLP, and MLR in predicting student job readiness reveals substantial differences. In a series of tests without using SMOTEENN, XGBoost consistently demonstrated superiority in accuracy, precision, recall, and F1-score compared with MLP and MLR. Notably, the XGBoost performance significantly surpasses the Multinomial Logistic Regression (MLR) model from previous research, which achieved an accuracy of only 53.9%. These results indicate that XGBoost has a better capability in classifying students regarding their job readiness, providing more accurate and reliable predictions across various scenarios. In the scenario with the highest accuracy, with a data split ratio of training to testing data at 9:1, XGBoost reaches an accuracy of 89%, precision of 84%, recall of 77%, and F1-score of 82%. In addition, SMOTEENN contributes to enhancing their performance although XGBoost remains superior in all situations.

This study also identifies several limitations that need to be considered. Limited data, small model size, and suboptimal preprocessing techniques are factors that limit the generalization of this research's findings. Therefore, further experiments with larger datasets, more complex models, and advanced preprocessing techniques are required to improve prediction quality in the future. Nevertheless, these findings provide valuable insights into the superiority of XGBoost in supporting decision-making regarding student job readiness, while SMOTEENN provides an additional boost in enhancing the performance of both predictions.

REFERENCES

[1] I. Kapareliotis, K. Voutsina, and A. Patsiotis, "Internship and employability prospects: assessing student's work readiness," High. Educ. Ski. Work. Learn., vol. 9, no. 4, pp. 538–549, 2019, doi: 10.1108/HESWBL-08-2018-0086.

[2] J. Winterton and J. J. Turner, "Preparing graduates for work readiness: an overview and agenda," *Education and Training*, vol. 61, no. 5, pp. 536–551, Jul. 2019, doi: 10.1108/ET-03-2019-0044.

[3] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, IEEE, pp. 1–6, Aug. 2018, doi: 10.1109/ICCUBEA.2018.8697857.

[4] A. Jain, K. shah, P. Chaturvedi dan A. Tambe, "Prediction and Analysis of Student Performance using Hybrid Model of Multilayer Perceptron and Random Forest," in *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, IEEE, pp. 1–7, Dec. 2018 doi: 10.1109/ICACAT.2018.8933580.

[5] H. Q. Nguyen, D. Dang, K. Nguyen, T. D. Le, A. Mai, and K. T. Huynh, "Career path prediction using XGBoost Model and students' academic results," *CTU Journal of Innovation and Sustainable Development*, vol. 15, pp. 62–75, 2023, doi: 10.22144/ctujoisd.2023.036.

[6] H. A. Salka and K. M. Lhaksmana, "Work readiness prediction of Telkom University students using multinomial logistic regression and random forest method," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *6*(4), 2022, https://doi.org/10.30865/mib.v6i4.4546

[7] A. Iqbal and S. Aftab, "A classification framework for software defect prediction using multi-filter feature selection technique and MLP," *International Journal of Modern Education and Computer Science*, vol. 12, no. 1, pp. 18–25, 2020, doi: 10.5815/ijmecs.2020.01.03.

[8] D. Aggarwal, S. Mittal, and V. Bali, "Prediction model for classifying students based on performance using machine learning techniques," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2 Special Issue 7, pp. 496–503, Jul. 2019, doi: 10.35940/ijrte.B1093.0782S719.

[9] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3360–3379, 2023, doi: 10.1080/10494820.2021.1928235.

[10] T. Hamim, F. Benabbou, and N. Sael, "Student profile modeling using boosting algorithms," *International Journal of Web-Based Learning and Teaching Technologies*, vol. 17, no. 5, pp. 1–13, Aug. 2021, doi: 10.4018/IJWLTT.20220901.oa4.

[11] M. Lin et al., "Detection of Ionospheric scintillation based on XGBoost model improved by SMOTE-ENN technique," 2021, doi: 10.3390/rs13132577.

[12] S. Chen, E. Xie, C. Ge, R. Chen, D. Liang, and P. Luo, "CycleMLP: A MLP-like Architecture for Dense Prediction," Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.10224

[13] Y. Qiu, J. Zhou, M. Khandelwal, H. Yang, P. Yang, and C. Li, "Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration," *Eng Comput*, vol. 38, pp. 4145–4162, Dec. 2022, doi: 10.1007/s00366-021-01393-9.

[14] H, Iqbal. "Implementing extreme gradient boosting (xgboost) classifier to improve customer churn prediction", 2020.

[15] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2018, doi: 10.1016/j.aci.2018.08.003.

[16] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "A Machine Learning Based Classification Method for Customer Experience Survey Analysis," *Technologies (Basel)*, vol. 8, no. 4, pp. 76, Dec. 2020, doi: 10.3390/technologies8040076.

[17] I. Rallis, I. Markoulidakis, I. Georgoulas, and G. Kopsiaftis, "A novel classification method for customer experience survey analysis," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jun. 2020, pp. 486–494. doi: 10.1145/3389189.3397999.

[18] E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 1: Literature review," *Remote Sensing*, vol. 13, no. 13. MDPI AG, Jul. 01, 2021. doi: 10.3390/rs13132450.