Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Application of the XGBoost Model with Hyperparameter Tuning for Industry Classification for Job Applicants

Akhmal Angga Syahputra[1)*] , Rujianto Eko Saputro[2]
[1]Informatics, Faculty of Computer Science, University of Amikom Purwokerto
[2]Information Technology, Faculty of Computer Science, University of Amikom Purwokerto
[1)]21sa29@mhs.amikompurwokerto.ac.id, [2)]rujianto@amikompurwokerto.ac.id,

Submitted : Jul 8, 2024 | **Accepted** : Jul 26, 2024 | **Published** : Jul 31, 2024

Abstract: The development of technology and changes in job market dynamics have created new challenges in aligning education with industry needs. In this research, the XGBoost model with hyperparameter tuning was applied for industry classification on job applicant data taken from the Kaggle dataset LinkedIn Job Postings in 2023. This dataset consists of 23 attributes with a total of 33,085 job vacancy data points. The experimental results show that both the model without hyperparameter tuning and with GridSearchCV produce the same classification accuracy, which is 0.89 or 89%, with stable precision, recall, and F1-Score values. The best parameters found in this study are colsample_bytree = 1.0, learning_rate = 0.3, max_depth = 6, min_child_weight = 1, n_estimators = 100, and subsample = 1.0. However, cross-validation using k-fold shows a significant increase in accuracy to 0.90, or 90%. This finding confirms that the use of cross-validation can improve the performance estimation of the model more accurately and robustly by utilizing all available data for training and testing. Moreover, the implementation of cross-validation demonstrates the importance of leveraging all data points to enhance model reliability and robustness. Future research can explore alternative hyperparameter tuning methods and apply the model to larger datasets to further validate the generalizability and reliability of the XGBoost model in different application contexts. Thus, this study underscores the significance of rigorous model evaluation techniques in achieving high-performing machine learning models.

Keywords: XGBoost; Hyperparameter tuning; GridSearchCV; Cross-validation; Industry classification

INTRODUCTION

The development of technology and changes in job market dynamics have created new challenges in aligning education with industry needs. One effort to address this is by improving vocational education effectiveness, especially in Technical and Vocational High Schools (SMK) in the technology and industry fields. However, SMK still faces complex issues, particularly regarding graduate professionalism and job opportunities or further education (Akbar et al., 2023; Lampiran & Smkn, n.d.) To tackle this challenge, the Ministry of Manpower (Kemnaker) has implemented the transformation policy for Vocational Training Centers (BLK) since 2021. This program follows the 6R concept:

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

institutional reform, training substance redesign, human resource revolution, facility revitalization, BLK rebranding, and relationship building. One substantial change is the integration of placement and training processes, including strengthening career counseling (Ministry of Manpower, 2021).

In this context, the right industry classification for job applicants becomes very important. This can help in aligning applicants' skills with industry needs, as well as provide better guidance in training and career development. To achieve this goal, the use of machine learning techniques such as XGBoost (Extreme Gradient Boosting) can be an effective solution. XGBoost is a classification model that excels in terms of computing speed and accuracy. The advantages of XGBoost include the ability to perform parallel computing automatically, support for customization of objective and evaluation functions, and better performance on various datasets (Farhan & Setiaji, 2023). Several recent studies have shown that XGBoost has a lower error score compared to other learning methods (Azis & Lestari, 2023). The use of hyperparameter tuning in XGBoost model optimization has been proven to significantly improve model performance (Punuri et al., 2023) (Ryan Afrizal et al., 2021). Techniques such as grid search, random search, and Bayesian optimization have been used to search for optimal combinations of hyperparameters (Probst et al., 2019). Recent research shows that effective hyperparameter tuning can substantially improve the accuracy of XGBoost models across various datasets (Alifah et al., 2024) (Akbar et al., 2023).

Based on the explanations and several previous research references that have been provided, XGBoost has the lowest error rate compared to other learning methods Therefore, in this study, we will apply the concept of XGBoost method for industry classification of job applicants, with the hope of achieving more accurate and effective results in aligning applicants' skills with industry needs

LITERATURE REVIEW

The application of XGBoost in various fields has shown its effectiveness in classification tasks. In the context of cybersecurity, XGBoost has managed to improve the accuracy of SYN attack detection to 96%, showing a significant improvement compared to other methods (Gunawan et al., 2022). In addition, XGBoost is used to classify malware with high efficiency, even on limited computing resources (Kumar & Geetha, 2020). In insurance, it was found that XGBoost is reliable in predicting motor vehicle insurance claims with a high level of accuracy and precision, achieving a prediction accuracy of 91% (Firdaus et al., 2021). Research also revealed that the use of XGBoost and particle swarm optimization in the classification of Parkinson's disease resulted in a significant increase in classification accuracy, with an accuracy of up to 94% (Kurnia et al., 2023).

Furthermore, the application of XGBoost for forest fire classification effectively improves environmental risk management (Muslim Karo Karo, 2020). The study also proposes the integration of XGBoost with SMOTE to improve intrusion detection in IoT environments, demonstrating the adaptability of XGBoost to unbalanced datasets that commonly occur in industrial IoT scenarios (Doghramachi & Ameen, 2023). The application of XGBoost in analyzing credit risk in an effort to optimize the customer screening process obtained classification results in a prediction model with an accuracy of 83%, so that it is included in the very good model value (80–89%) (Saputra et al., 2024). This proving the effectiveness of XGBoost in improving the performance of the recommendation system and demonstrating the flexibility and capability of this method in various application domains (Ris et al., 2019).

METHOD

Research Flow

In the study, there are seven main stages used: data collection, data processing, splitting data, cross-validation, modeling, hyperparameter tuning, and evaluation. Several stages must be carried out in stages, including data collection and data preprocessing, before proceeding to splitting the data, which can be seen in Figure 1.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

SinkrOn
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

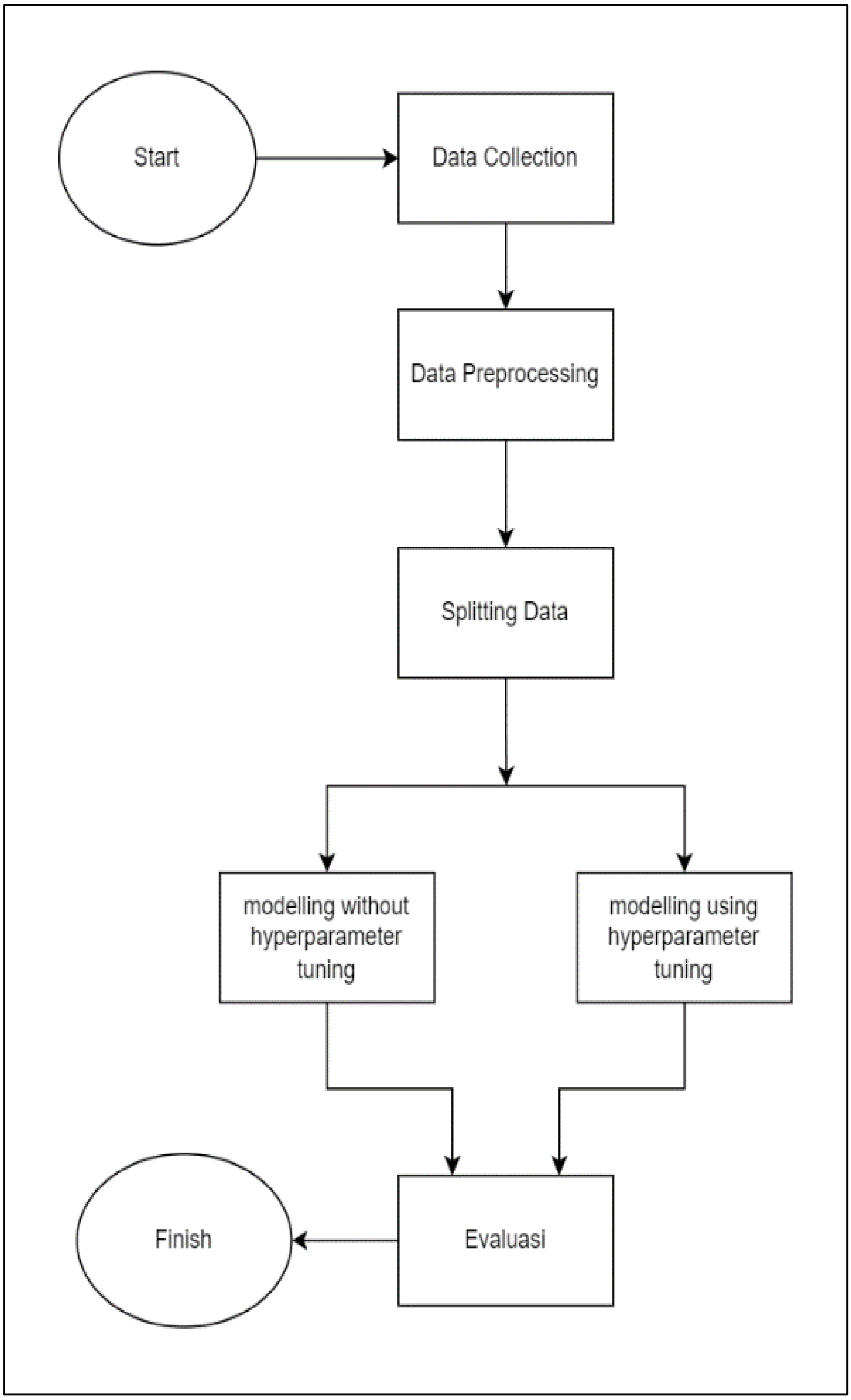

Figure 1 Research stage.

Data Collection

In this study, the data used came from the public dataset "LinkedIn Job Postings" available on Kaggle in 2023. This dataset includes various information about job vacancies, job details, company details, and mapping. This dataset consists of 4 folders and 11 CSV files, but in the research, we used 23 attributes with a total of 33,085 job vacancy data. This data will be used to evaluate and develop an accurate industry recommendation model for job applicants, with a focus on classification to capture different aspects of each available job. Table 1 shows the datasets used.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 1 Dataset Information

Attributes	Type
Job Id	Integer
Company Id	Float
heading	String
Maximum salary	Float
Middle Salary	Float
Minimum Salary	Float
Payment Period	String
Types of Formatted Jobs	String
Application	Float
Remote allowed	Float
Job type	String
Currency	String
Compensation Type	String
Industry ID	Float
Skill	String
Industry name	String
Company Name	String
Country	String
Number of Employees	Float

Data Preprocessing

Data preprocessing is the initial stage in data preparation, which includes cleaning, handling lost data, and adapting the raw data to fit the format required for subsequent analysis. This stage of preprocessing includes collecting, cleaning, and labeling. This process must be prepared optimally; otherwise, it will result in inaccurate analyses and misleading conclusions (Muslim Karo Karo, 2020)

Data cleaning is very important for analyzing data for industry classification processes using machine learning algorithms. Data cleaning includes handling missing data, deleting duplicate data, and formatting data that is not appropriate. This can remove incomplete data or use mean or mode imputation to manage missing data (Alhamad et al., 2019). Furthermore, dimension reduction is a data analysis technique to reduce the number of dimensions or attributes in a dataset. Outliers can be detected using a boxplot, where data values greater than Q3 + 15 * IQR or less than Q1 - 15 * IQR are considered outliers, where Q1 is the lower quartile, Q3 is the upper quartile, and IQR is the interquartile range (Q3 - Q1) (Syukron et al., 2020). Converting categorical variables to numeric variables can be done using techniques such as one-hot encoding, label encoding, or manual mapping for ordinal variables (Hamami & Dahlan, 2022).

After grouping the target attribute classes, data balance checking was performed the use of SMOTE and near miss under-sampling methods was employed to balance the sample sizes across classes, which proved to enhance model performance (Sabilla & Bella Vista, 2021). Feature engineering, the process of creating data representations that improve model effectiveness, was also applied to ensure optimal performance (Kuhn & Johnson, 2019). The results of the imbalance stage in this study can be seen in Figure 2 and Figure 3.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

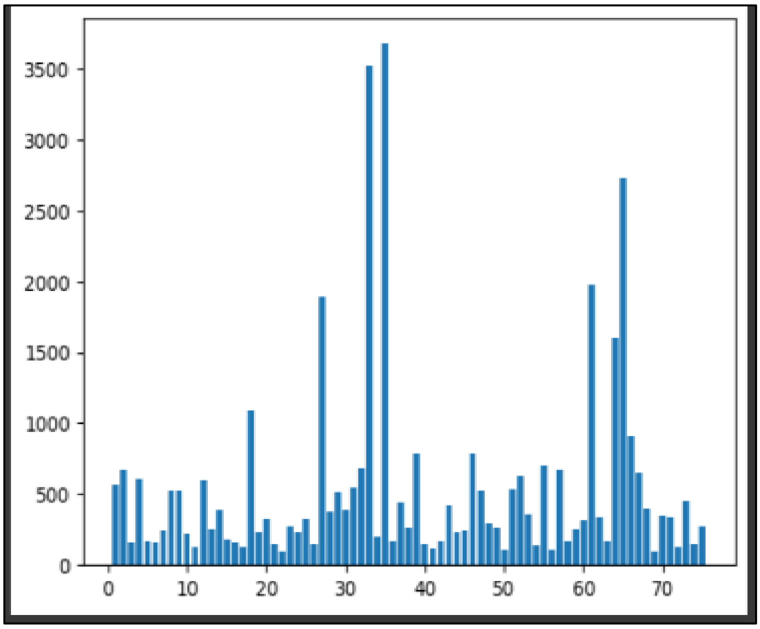

Figure 2 Before Imbalance.

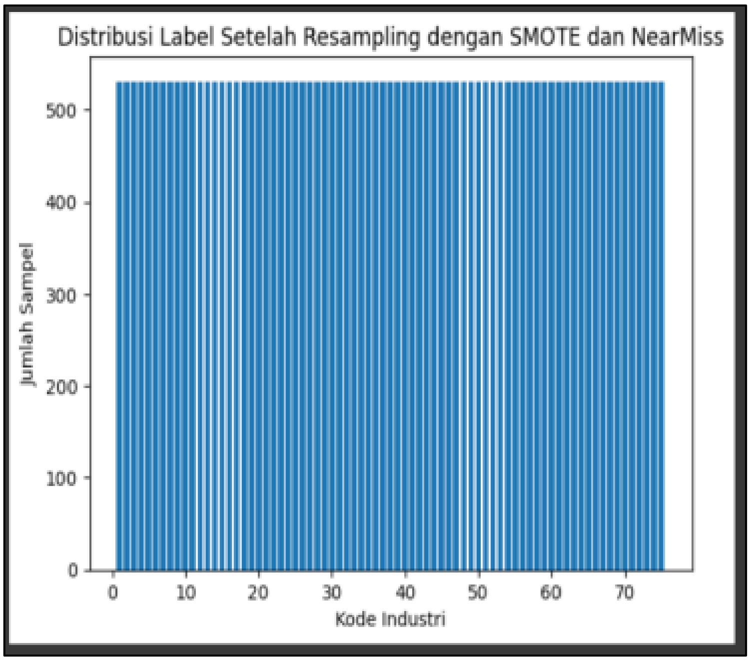

Figure 3 After Imbalance

Splitting data

The next stage is the separation of data in the data frame, which is divided into testing data and training data with a separation ratio of 80:20. 80% of the entire data frame is used for training data, and 20% of the data frame is used for data testing or testing data.

Data Transformation

Perform data transformation, such as normalization or standardization. In this case, use the standard scaler method, which can help improve the performance of machine learning models (Reza & Muhammad Syaifur Rohman, 2024). The formula for performing standard scaling:

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$Z = \frac{x-\mu}{\sigma} \quad (1)$$

Z: value of the newly scaled data
x: original value of the data
μ: mean of the data
σ: deviation of the data

Modeling

XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is an effective machine learning algorithm model for classification tasks (Reza & Muhammad Syaifur Rohman, 2024). The upgrade method uses a new model to predict errors from the previous model. The addition of this model is carried out until there are no more significant error corrections (Fatihah et al., 2024). Using a new regularization technique to control overfitting makes XGBoost different from other gradient enhancements (Akbar et al., 2023). Overfitting is a condition in which the data used for training is the "best" data, but its accuracy will be reduced if the test is performed with other data (Kuhn & Johnson, 2019). To reduce errors when creating new models, this algorithm uses gradient decline. The term gradient increase is also used by this algorithm (Ihsan et al., 2024).

$$\hat{y}_i^t = \Sigma_{k=1}^{t} f_k(X_i) \quad (2)$$

Information:
$\hat{y}_i^t$ = Final tree model
$f_k(X_i)$= New model built
t = The total number of models from the base tree models

Hyper Parameter Tuning

In this study, two methods were used: XGBoost without hyperparameter tuning, which means all parameters used followed the default settings The second method involved hyperparameter tuning using GridSearchCV, a method to find the best set of hyperparameters from a grid of values (Kohli & Joshi, 2021). Hyperparameter adjustment was performed using the training validation data set. The data partitioning variation was also influenced by this optimization (Putri et al., 2024). It's important to remember that the first challenge is finding the ideal hyperparameter values. However, manual exploration requires a lot of resources and time. GridSearchCV automates the hyperparameter tuning process, making it easier and allowing for systematic search across different values to find the best configuration (Cahyani et al., 2024). After obtaining the best parameters, evaluate the model's performance using cross-validation (Rusdah & Murfi, 2020).

Evaluation

The final stage of this research is evaluation. Model evaluation aims to determine the accuracy of the machine learning model predictions (Sirait et al., 2024). The confusion matrix has several evaluation components: TP (true positive) for correctly predicted positive data, TN (true negative) for correctly predicted negative data, FP (false positive) for negative data predicted as positive, and FN (false negative) for positive data predicted as negative (Wicaksono et al., 2024). The evaluation results are obtained by calculating accuracy, macro-average precision, recall, and F1-score (Darvishy et al., 2020). The performance metric formulas are as follows:

Accuracy is used to describe the ability of the model to classify data correctly (Abdurrahman et al., 2022).

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

$$Accuracy = \frac{TP+TN}{TP + FP + TN + FN} \quad (3)$$

Precision is a metric that indicates the proportion of true positive predictions.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Recall is a metric that measures the number of positive cases correctly identified by a prediction

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

The F1-Score is an evaluation measure that integrates the values of precision and recall to provide an overall view of the model's performance.

$$F1\text{- }score = 2 \times \frac{precision \times recall}{precision + recall} \quad (6)$$

Macro-averaged precision and recall are the mechanisms of the user's precision and acquisition values, respectively; they are calculated individually for each user as specified in, respectively (Darvishy et al., 2020).

$$Macro\text{-}Averaged\text{-}precision = \frac{\Sigma_i^n{=}_1 precision_i}{n} \quad (7)$$

$$Macro\text{-}Averaged\text{-}recall = \frac{\Sigma_i^n{=}_1 recall_i}{n} \quad (8)$$

The Macro-averaged F1 score is the harmonic mean of the average macro precision and the average macro recall.

$$Macro\text{-}Averaged\text{-}F1\text{-}Score = 2 \times \frac{Macro-Averaged-precision \times Macro-Averaged-recall}{Macro-Averaged-precision + Macro-Averaged-recall} \quad (9)$$

RESULT

This chapter describes the results obtained from each experimental stage conducted in the research. The experiments were carried out without using hyperparameters, using grid search, or adding k-fold.

Experimental Results Without Hyperparameter Tuning

Based on the implementation that has been carried out, the following results of experiments carried out without tuning hyperparameters or using by default, are shown in the following table 2:

Table 2 Classification Report Xgboost without HP

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.89	0.96	0.91	0.93

The accuracy result from the experiment without hyperparameter tuning on the job postings dataset yielded an accuracy score of 0.89 or 89%

Results of Hyperparameter Tuning Experiment

Based on its implementation, using gridsearch hyperparameter tuning with parameters learning_rate = 001, 01, and 03, max_depth = 5, 6, and 7, n_estimators = 50, 100, and 200, subsample = 08 and 10, colsample_bytree = 08 and 10, and min_child_weight = 1 and 5, The best parameters for the experiment are colsample_bytree = 10, learning_rate = 03, max_depth = 6, min_child_weight = 1, n_estimators =

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

100, and subsample = 10. The results of the industry classification using gridsearch are displayed in the following table:

Table 3 Classification Report using HP

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.89	0.96	0.91	0.93

The accuracy result from the hyperparameter tuning experiment using gridsearch on the job postings dataset yielded an accuracy score of 0.89, or 89%.

Best parameter evaluation results with cross validation

Based on its implementation, this method uses k-fold cross-validation with 10 folds, or k-fold = 10. The results are displayed in the following table:

Table 4

Evaluation of Best Parameters with Cross-Validation

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.90	0.96	0.91	0.93

The accuracy result from the best parameter evaluation experiment using cross-validation yielded an accuracy score of 0.90 or 90%.

In previous research, the tree-based classification model categorized BMI into six index classes, and the results showed that the XGBoost classification model had the highest accuracy after hyperparameter tuning, with a test data accuracy of 83,7% (Alifah et al., 2024). This ongoing study indicates higher results, with an accuracy reaching 90% after evaluating the best parameters using cross-validation. These findings reaffirm the importance of using hyperparameter tuning methods to enhance model performance.

DISCUSSIONS

The test results show that the implementation of hyperparameter tuning using GridSearch initially did not have a significant impact on the model's performance. However, after evaluation using cross-validation, it was found that the use of the best parameter improved the performance of the model. In comparison, experiments without the application of hyperparameters resulted in a classification accuracy of 0.89, or 89%.

The test results show that the implementation of hyperparameter tuning using GridSearch initially did not have a significant impact on the model's performance. The experimental results show that using GridSearch for hyperparameter tuning results in comparable performance to experiments without tuning. The best parameters found were colsample_bytree = 1.0, learning_rate = 0.3, max_depth = 6, min_child_weight = 1, n_estimators = 100, and subsample = 1.0. With these parameters, the classification accuracy remains 0.89, or 89%, and the precision, recall, and F1 scores remain the same as in experiments without hyperparameter tuning. However, significant differences were seen when the model was evaluated using k-fold cross-validation with k = 10. The accuracy of the model increases to 0.90 or 90% after applying cross-validation. This improvement suggests that cross-validation can help assess model performance more accurately and robustly by using all available data for training and testing. However, after evaluation using cross-validation, it was found that the use of the best parameter improved the performance of the model. In comparison, experiments without the application of hyperparameters resulted in a classification accuracy of 0.89, or 89%.

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

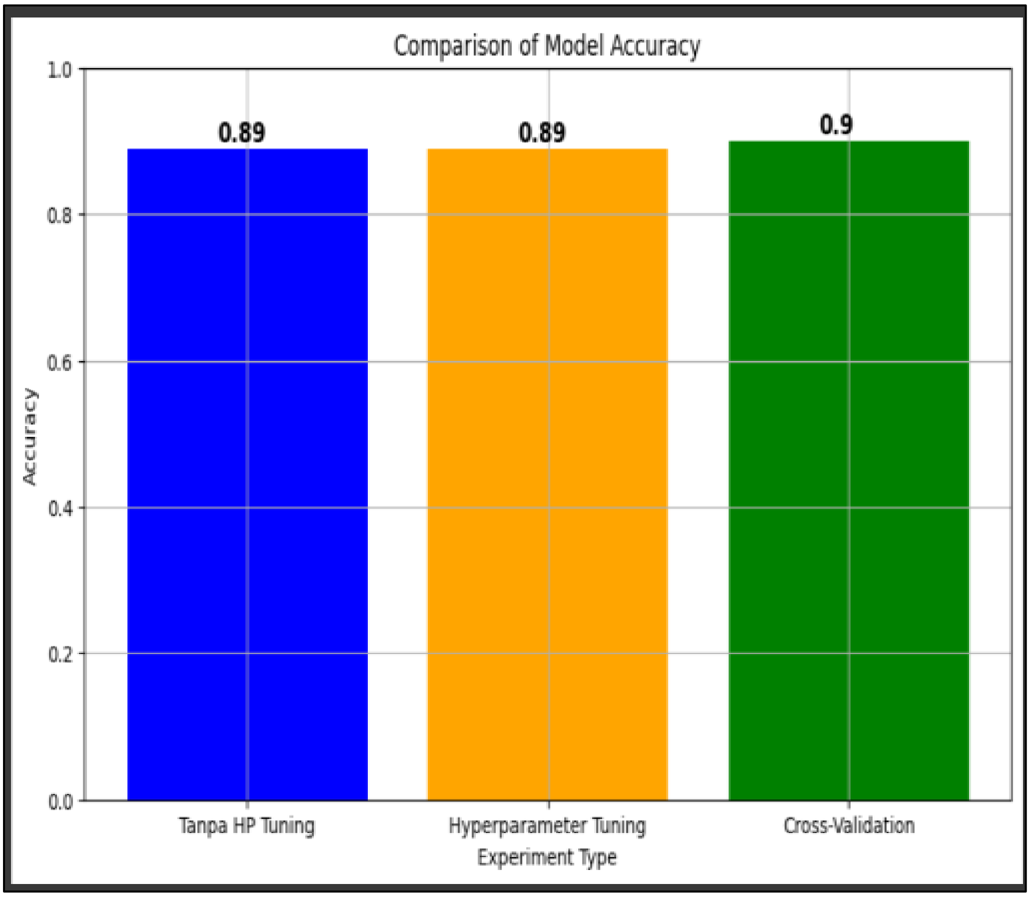

Figure 4 XGBoost Accuracy Comparison

The experiment results show that tuning hyperparameters doesn't directly improve the performance of the model in job applicant industry classification using XGBoost. However, applying cross-validation to evaluate the best parameters can produce more reliable performance estimates. This finding emphasizes the importance of using cross-validation as a tool to validate and enhance the reliability of model performance estimates, even though there is no direct improvement from tuning hyperparameters.

CONCLUSION

Based on research conducted by implementing the XGBoost model with hyperparameter tuning for industry classification on job applicant data, The experimental results show that both the model without hyperparameter tuning and with gridsearch produce the same classification accuracy, which is 0.89 or 89%, with stable precision, recall, and F1-Score values. However, evaluation using k-fold cross-validation reveals a significant improvement in accuracy to 090, or 90%. This finding confirms that the use of cross-validation can enhance the estimation of model performance more accurately and robustly, even though hyperparameter tuning did not show a noticeable difference in this study. Recommendations for further research include exploring alternative methods for hyperparameter tuning and evaluating models on a larger dataset to deepen understanding of model generalizability and reliability in various practical contexts.

ACKNOWLEDGMENT

This research is an output of the Merdeka Belajar Kampus Merdeka (MBKM) programme and the research funding is fully borne by the Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Univesitas Amikom Purwokerto.

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2024
DOI : https://doi.org/10.33395/sinkron.v8i3.13840
e-ISSN : 2541-2019
p-ISSN : 2541-044X

REFERENCES

Abdurrahman, G., Oktavianto, H., & Sintawati, M. (2022). Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridesearch dan Random Search Pada Klasifikasi Penyakit Diabetes. *INFORMAL: Informatics Journal*, *7*(3), 193. https://doi.org/10.19184/isj.v7i3.35441

Akbar, M., Nadyanto, W., & Komalasari, D. (2023). *Prediction of Total Transaction Using Extreme Gradient Boosting (XGBoost)*. *9*(1), 30–45.

Alhamad, A., Azis, A. I. S., Santoso, B., & Taliki, S. (2019). Prediksi Penyakit Jantung Menggunakan Metode-Metode Machine Learning Berbasis Ensemble – Weighted Vote. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *5*(3), 352. https://doi.org/10.26418/jp.v5i3.37188

Alifah, R. N., Najib, M. K., Nurdiati, S., Sari, A. P., Herlambang, K., Putri, T., Ginting, B., & Sya'adah, S. N. (2024). Perbandingan Metode Tree Based Classification untuk Masalah Klasifikasi Data Body Mass Index. *Indones. J. Math. Nat. Sci*, *47*(1), 2024. https://journal.unnes.ac.id/journals/JM/index

Azis, A. M. H., & Lestari, D. P. (2023). XGBoost and Convolutional Neural Network Classification Models on Pronunciation of Hijaiyah Letters According to Sanad. *Jurnal Online Informatika*, *8*(2), 194–203. https://doi.org/10.15575/join.v8i2.1081

Cahyani, I. A., Ashuri, P. I., & Aditya, C. S. K. (2024). Stunting Disease Classification Using Multi-Layer Perceptron Algorithm with GridSearchCV. *Sinkron*, *9*(1), 392–401. https://doi.org/10.33395/sinkron.v9i1.13245

Darvishy, A., Ibrahim, H., Sidi, F., & Mustapha, A. (2020). HYPNER: A Hybrid Approach for Personalized News Recommendation. *IEEE Access*, *8*, 46877–46894. https://doi.org/10.1109/ACCESS.2020.2978505

Doghramachi, D. F., & Ameen, S. Y. (2023). Internet of Things (IoT) Security Enhancement Using XGboost Machine Learning Techniques. *Computers, Materials and Continua*, *77*(1), 717–732. https://doi.org/10.32604/cmc.2023.041186

Farhan, N. M., & Setiaji, B. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, *12*(2), 284–301. http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135

Fatihah, A. M., Dharmawan, K., & Swastika, P. V. (2024). *Implementation of X-Gradient Boosting in Banking Stock Price Predictions* (Issue Icamsac 2023). Atlantis Press International BV. https://doi.org/10.2991/978-94-6463-413-6_17

Firdaus, A. A., Komarudin, A., Statistika, M. P., Matematika, F., Ilmu, D., & Alam, P. (2021). Klasifikasi Pemegang Polis Menggunakan Metode XGBoost. *Prosiding Statistika*, *7*(2), 704–710. http://dx.doi.org/10.29313/.v0i0.30320

Gunawan, R. G., Erik Suanda Handika, & Edi Ismanto. (2022). Pendekatan Machine Learning Dengan Menggunakan Algoritma Xgboost (Extreme Gradient Boosting) Untuk Peningkatan Kinerja Klasifikasi Serangan Syn. *Jurnal CoSciTech (Computer Science and Information Technology)*, *3*(3), 453–463. https://doi.org/10.37859/coscitech.v3i3.4356

Hamami, F., & Dahlan, I. A. (2022). Klasifikasi Cuaca Provinsi Dki Jakarta Menggunakan Algoritma Random Forest Dengan Teknik Oversampling. *Jurnal Teknoinfo*, *16*(1), 87. https://doi.org/10.33365/jti.v16i1.1533

*name of corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Ihsan, C. N., Agustina, N., Naseer, M., Gusdevi, H., Rusdi, J. F., Hadhiwibowo, A., & Abdullah, F. (2024). Comparison of Machine Learning Algorithms in Detecting Tea Leaf Diseases. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *8*(1), 135–141. https://doi.org/10.29207/resti.v8i1.5587

Kohli, S., & Joshi, P. (2021). *" A Brief Study on Random Forest Using Python ." 3*(6), 2063–2069. https://doi.org/10.35629/5252-030620632069

Kuhn, M., & Johnson, K. (2019). Feature Engineering and Selection. In *Feature Engineering and Selection*. CRC Press. https://doi.org/10.1201/9781315108230

Kumar, R., & Geetha, S. (2020). Malware classification using XGboost-Gradient boosted decision tree. *Advances in Science, Technology and Engineering Systems*, *5*(5), 536–549. https://doi.org/10.25046/AJ050566

Kurnia, D., Itqan Mazdadi, M., Kartini, D., Adi Nugroho, R., & Abadi, F. (2023). Seleksi Fitur dengan Particle Swarm Optimization pada Klasifikasi Penyakit Parkinson Menggunakan XGBoost. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *10*(5), 1083–1094. https://doi.org/10.25126/jtiik.20231057252

Lampiran, L., & Smkn, M. O. U. (n.d.). *library.uns.ac.id digilib.uns.ac.id*.

Muslim Karo Karo, I. (2020). Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan. *Journal of Software Engineering, Information and Communication Technology*, *1*(1), 11–18.

Probst, P., Boulesteix, A. L., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, *20*, 1–32.

Punuri, S. B., Kuanar, S. K., Kolhar, M., Mishra, T. K., Alameen, A., Mohapatra, H., & Mishra, S. R. (2023). Efficient Net-XGBoost: An Implementation for Facial Emotion Recognition Using Transfer Learning. *Mathematics*, *11*(3), 1–24. https://doi.org/10.3390/math11030776

Putri, R. T. E., Zeniarja, J., Winarno, S., Cahyani, A. N., & Maulani, A. A. (2024). GridSearch and Data Splitting for Effectiveness Heart Disease Classification. *Sinkron*, *9*(1), 317–331. https://doi.org/10.33395/sinkron.v9i1.13198

Reza, A. A. R., & Muhammad Syaifur Rohman. (2024). Prediction Stunting Analysis Using Random Forest Algorithm and Random Search Optimization. *Journal of Informatics and Telecommunication Engineering*, *7*(2), 534–544. https://doi.org/10.31289/jite.v7i2.10628

Ris, L., Parc, A., Fernanda, M., & Wanderley, B. (2019). *O Ptimization for F Eature S Election in Dna*. *11*(May), 1–17.

Rusdah, D. A., & Murfi, H. (2020). XGBoost in handling missing values for life insurance risk prediction. *SN Applied Sciences*, *2*(8), 1–10. https://doi.org/10.1007/s42452-020-3128-y

Ryan Afrizal, M., Adi Nugroho, R., Kartini, D., Herteno, R., Ahmad Yani Km, J., & Selatan, K. (2021). Xgboost Dengan Random Search Hyper-Parameter Tuning Untuk Klasifikasi Situs Phising. *Jurnal Ilmu Komputer*, *15*(1), 40–47.

Sabilla, W. I., & Bella Vista, C. (2021). Implementasi SMOTE dan Under Sampling pada Imbalanced Dataset untuk Prediksi Kebangkrutan Perusahaan. *Jurnal Komputer Terapan*, *7*(2), 329–339.

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

https://doi.org/10.35143/jkt.v7i2.5027

Saputra, A. A., Sari, B. N., Rozikin, C., Singaperbangsa, U., & Abstrak, K. (2024). Penerapan Algoritma Extreme Gradient Boosting (Xgboost) Untuk Analisis Risiko Kredit. *Jurnal Ilmiah Wahana Pendidikan*, *10*(7), 27–36. https://doi.org/10.5281/zenodo.10960080

Sirait, E. M., Silalahi, R., Tambunan, A. A., & Amalia, J. (2024). News Classification Using Bidirectional Long Short Term Memory and GloVe. *Sinkron*, *9*(1), 112–124. https://doi.org/10.33395/sinkron.v9i1.13005

Syukron, M., Santoso, R., & Widiharih, T. (2020). Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data. *Jurnal Gaussian*, *9*(3), 227–236. https://doi.org/10.14710/j.gauss.v9i3.28915

Wicaksono, D. F., Basuki, R. S., & Setiawan, D. (2024). Peningkatan Performa Model Machine Learning XGBoost Classifier melalui Teknik Oversampling dalam Prediksi Penyakit AIDS. *Jurnal Media Informatika Budidarma*, *8*(2), 736–747. https://doi.org/10.30865/mib.v8i2.7501

*name of corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.