XGBOOST HYPERPARAMETER OPTIMIZATION USING RANDOMIZEDSEARCHCV FOR ACCURATE FOREST FIRE DROUGHT CONDITION PREDICTION

Nur Alamsyah[1*]; Budiman[2]; Titan Parama Yoga[3]; R Yadi Rakhman Alamsyah[4]

Information Systems[1, 3], Informatics[2, 4]
Universitas Informatika Dan Bisnis Indonesia, Bandung, Indonesia[1, 2, 3, 4]
http://www.unibi.ac.id[1, 2, 3, 4]
nuralamsyah@unibi.ac.id[1*], budiman@unibi.ac.id[2], titanparamayoga@gmail.com[3], r.yadi@unibi.ac.id[4]
(*) Corresponding Author

Abstract— *Climate change and increasing global temperatures have increased the frequency and intensity of forest fires, making fire risk evaluation increasingly important. This study aims to improve the accuracy of predicting forest fuel drought conditions (Drought Code) by using the XGBoost algorithm optimized with RandomizedSearchCV. The research methods include collecting data related to forest fires, preprocessing data to ensure quality and consistency, and using RandomizedSearchCV for XGBoost hyperparameter optimization. The results showed that the optimized XGBoost model resulted in a decrease in Mean Squared Error (MSE) and an increase in R-squared value compared to the default model. The optimized model achieved an MSE of 0.0210 and R^2 of 0.9820 on the test data, indicating significantly improved prediction accuracy for forest fuel drought conditions. These findings emphasize the importance of hyperparameter optimization in improving the accuracy of predictive models for forest fire risk assessment.*

Keywords: *forest fire prediction, hyperparameter optimization, XGBoost.*

Abstrak— *Perubahan iklim dan peningkatan suhu global telah meningkatkan frekuensi dan intensitas kebakaran hutan, sehingga evaluasi risiko kebakaran menjadi semakin penting. Penelitian ini bertujuan untuk meningkatkan akurasi prediksi kondisi kekeringan bahan bakar hutan (Drought Code) dengan menggunakan algoritma XGBoost yang dioptimalkan dengan RandomizedSearchCV. Metode penelitian meliputi pengumpulan data terkait kebakaran hutan, preprocessing data untuk memastikan kualitas dan konsistensi, dan menggunakan RandomizedSearchCV untuk optimasi hyperparameter XGBoost. Hasil penelitian menunjukkan bahwa model XGBoost yang telah dioptimasi menghasilkan penurunan Mean Squared Error (MSE) dan peningkatan nilai R-squared dibandingkan dengan model default. Model yang dioptimasi mencapai MSE sebesar 0.0210 dan R^2 sebesar 0.9820 pada data uji, yang mengindikasikan peningkatan akurasi prediksi secara signifikan terhadap kondisi kekeringan bahan bakar hutan. Temuan ini menekankan pentingnya optimasi hyperparameter dalam meningkatkan akurasi model prediktif untuk penilaian risiko kebakaran hutan.*

Kata Kunci*: prediksi kebakaran hutan, optimasi hyperparameter, XGBoost.*

INTRODUCTION

Climate change and increasing global temperatures have caused an increase in the frequency and the intensity of forest fires around the world (Jones et al., 2022). According to research by (Korená Hillayová et al., 2023), "Climate change significantly increases the risk of wildfires, which can result in large ecological and economic losses". Accurate fire risk assessment is becoming increasingly important for effective mitigation and response to wildfires (Carta et al., 2023). In this context, the prediction of forest fuel drought conditions, as measured by the Drought Code (DC), plays an important role in fire risk assessment. The XGBoost algorithm has been extensively used in various predictive applications due to its ability to handle complex data and produce accurate predictions (Putrada et al., 2022). However, the performance of XGBoost is highly dependent on the proper selection of hyperparameters, which can significant impact the prediction accuracy (Wu et al., 2023). The study is to improve the accuracy of predicting forest fuel drought conditions using XGBoost that is optimized with

RandomizedSearchCV. The methods used include collecting data related to forest fires, preprocessing data to ensure quality and consistency, and using RandomizedSearchCV for XGBoost hyperparameter optimization.

Several previous studies have discussed the importance of hyperparameter optimization in improving the performance of predictive models. (Zhao et al., 2024). proposed RandomizedSearch as a more efficient method than GridSearch, as it allows a broader hyperparameter search with fewer iterations RandomizedSearchCV has been shown to provide competitive results with GridSearchCV with lower computation time (Ehsani & Hosseini, 2024). In a study by (Baseer et al., 2023), RandomizedSearchCV was used for hyperparameter optimization on the XGBoost model in retail sales prediction, which showed a significant improvement in model accuracy compared to the default parameters. In addition, the use of XGBoost for forest fire prediction has been discussed (Sun et al., 2024), where the model showed superior performance compared to other algorithms such as Random Forest and SVM.

According to (Alamsyah, Budiman, et al., 2023), “Hyperparameter optimization can significantly improve model performance, especially in big data applications”. This research shows that the combination of optimization techniques with advanced machine learning algorithms such as XGBoost can provide more accurate and reliable predictions. However, these studies have not specifically examined the use of RandomizedSearchCV for XGBoost hyperparameter optimization in the context of forest fuel drought condition prediction. Therefore, this study introduces a new approach by combining the power of XGBoost and the efficiency of RandomizedSearchCV to produce more accurate predictive models in forest fire risk assessment.

MATERIALS AND METHODS

The study was conducted to improve the accuracy of predicting forest fuel drought conditions (Drought Code) by using the XGBoost algorithm optimized through RandomizedSearchCV. The study started with data collection that included weather features, geographical locations, and dates relevant to forest fires. After data collection, preprocessing steps were performed to ensure data quality and consistency, including addressing missing values, encoding categorical data, and normalizing numerical features. The preprocessed data is then separated into a training set and a testing set with a ratio of 80:20 to ensure a fair evaluation of the model's performance (Alamsyah et al., 2024). The next step is hyperparameter optimization using RandomizedSearchCV, which allows finding the best hyperparameters in a predefined search space, using 5-fold cross-validation to validate the results.

The resulting model from hyperparameter optimization is evaluated using Mean Squared Error (MSE) and R-squared on test data, and cross-validation is performed to ensure that the model is not overfitting and can generalize well to new data (Alamsyah, Saparudin, et al., 2023). In addition, a learning curve visualization was used to monitor the model's performance during training and validation, providing additional insight into the model's ability to handle different data (Qin et al., 2024). Our proposed method is shown in Figure 1.

Data Collected

The data used in this study was sourced from the Kaggle portal, which provided datasets related to forest fires (THYGE PEDERSEN, 2023). This dataset consists of 517 data rows and includes various features relevant for the prediction of forest fuel drought conditions (Drought Code). The features used in this study include the X-coordinate and Y-coordinate of the fire location to define the geographical position of the fire. Time features include the month and day of the fire occurrence, providing temporal context to the data. Other features include the Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), and Drought Code (DC), which are the prediction targets of this study. In combination, the Initial Spread Index (ISI) was also used to understand the early fire dynamics. Weather features such as air temperature in degrees Celsius (temp), relative humidity in percent (RH), wind speed in km/h (wind), and rainfall in mm/m2 (rain) are also included to provide the meteorological context affecting the fire. Finally, the area burnt in hectares (area) feature was used to measure the impact of the fire. As an example, we provide the data values for each feature in Table 1.

This data includes important information needed to build an accurate predictive model to evaluate the risk of forest fuel drought. The data collection process is a crucial first step in ensuring that the model developed has a reliable and relevant database.

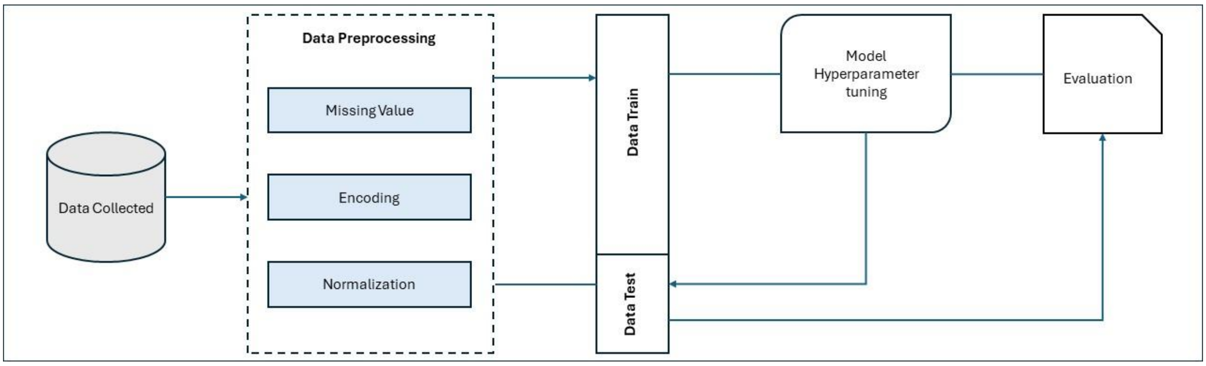
Data Preprocessing

The initial phase of data preprocessing involves addressing missing values to ensure that the dataset used is complete, thereby avoiding any potential impact on model performance (Erpurini et al., 2023). In this research, missing values are handled through forward filling to maintain data integrity. Subsequently, numerical feature normalization is performed to standardize all features to a consistent scale, thereby optimizing

Table 1. Sample Data from the Dataset

x	y	month	day	ffmc	dmc	dc	isi	temp	rh	wind	rain	area
7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0
7	4	oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0
7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0
8	6	mar	fri	91.7	33.3	77.5	9	8.3	97	4	0.2	0
8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0
8	6	aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0
8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0
8	6	aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0
8	6	sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0

Source: (THYGE PEDERSEN, 2023)



Source: (Research Result, 2024)

Figure 1. Proposed Method

model performance (Huang et al., 2023). The normalized features include Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), air temperature (temp), relative humidity (RH), wind speed (wind), rainfall (rain), and area burned (area). Normalization is done using StandardScaler which transforms the data so that it has a mean of 0 and a standard deviation of 1. Preprocessing this data is very crucial to ensure that the prediction model can work well and provide accurate results. By preprocessing, we can reduce the noise in the data and improve the quality of the features used for model training (Albahra et al., 2023).

Splitting Data

After the data preprocessing process is complete, the next step is to separate the data into a training set and a testing set. This separation is to ensure that the model can be evaluated fairly and not overfitting the training data. In this study, the data is separated with a ratio of 80:20, where 0.8 of the data is used to train the model and the remaining 0.2 is used to test the performance of the model.

Model Hyperparameter Tuning

The XGBoost algorithm aims to enhance predictive accuracy by iteratively improving an objective function. This objective function consists of two main components: the loss function and the regularization term (Todorovic et al., 2023). The loss function, denoted as $(\mathrm{l}(\mathrm{y_i}, \hat{y}_\mathrm{i}))$, measures the difference between the actual target values $((\mathrm{y_i}))$ and the predicted values $((\hat{y}_\mathrm{i}))$produced by the model. The goal is to minimize this loss, thereby improving the model's accuracy. In addition to the loss function, the objective function includes a regularization term, represented as $(\Omega(\mathrm{f_k}))$. The term quantifies the complexity of the model, which helps prevent overfitting by penalizing more complex models. Overfitting occurs when a model performs well on training data but poorly on unseen test data. By including this regularization term, XGBoost balances the model's fit on the training data and its ability to generalize to new data. Mathematically, the objective function of XGBoost can be expressed as:

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

Where (Θ) represents the set of hyperparameters that the model will optimize. The first summation, $(\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{l}(\mathrm{y_i}, \hat{y}_\mathrm{i}))$, aggregates the loss over all (n) data points. The second summation, $(\sum_{\mathrm{k}=1}^{\mathrm{K}} \Omega(\mathrm{f_k}))$, sums the regularization terms for all (K) trees in the ensemble. By minimizing this combined objective function, XGBoost ensures that the model not only fits the training data well but also maintains a level of simplicity that promotes better generalization to new data.

PILAR Nusa Mandiri: Journal of Computing and Information System
Vol. 20, No. 2 September 2024 | DOI: 10.33480/pilar.v20i2.5569

Evaluation

The evaluation of the XGBoost model, optimized using RandomizedSearchCV, involves assessing its performance on a separate test dataset that was not used during the training phase. This evaluation helps determine how well the model generalizes to new, unseen data. Two primary metrics are used for this evaluation: Mean Squared Error (MSE) and R-squared R^2. MSE calculates the average squared discrepancy between the actual target values (y_i) and and their corresponding predicted values $(\hat{y}_i)$. It is a commonly used metric for regression tasks because it penalizes larger errors more severely, providing a clear indication of the model's predictive accuracy. The formula for MSE is given by:

$$[\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2] \tag{2}$$

Where (n) represents the number of data points in the test set, (y_i) denotes the actual target value for the (i)-th data point, and $(\hat{y}_i)$ represents the predicted value for the (i)-th data point. A lower MSE value indicates better predictive performance, as it means the predicted values are closer to the actual values. R-squared (R^2), also referred to as the coefficient of determination, quantifies the amount of variation in the dependent variable that is explained by the independent variables. It serves as a gauge of how effectively the model's predictions align with the observed data. The formula for (R^2) is expressed as:

$$[R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}] \tag{3}$$

Where (y_i) denotes the actual target value for the (i)-th data point, $(\hat{y}_i)$represents the predicted value for the (i)-th data point, and $(\bar{y})$ is the mean of the actual target values. The (R^2) value ranges from 0 to 1. A higher value indicates that a significant portion of the variance in the target variable is accounted for by the model, while a lower value suggests that the model explains less of the variance in the target variable. After training the XGBoost model with the best hyperparameters obtained from RandomizedSearchCV, the model is evaluated on the test set using MSE and (R^2)metrics. The evaluation process involves predicting the target values for the test data and then calculating the MSE and (R^2) to assess the model's performance.

RESULTS AND DISCUSSION

The evaluation of the XGBoost model, both with default parameters and with hyperparameter tuning using RandomizedSearchCV, provides insights into the effectiveness of these approaches in predicting forest fire drought conditions. This section presents the results of the model evaluation, including a comparison of the performance metrics for both model configurations. The discussion focuses on interpreting these results, highlighting the impact of hyperparameter tuning on model performance, and considering the practical implications of these findings for forest management and risk assessment.

Result

Data preprocessing is an important step to ensure the quality and consistency of data before it is used in a prediction model. In this study, the dataset used contains information about forest fires with various features such as geographic location (x, y), month (month), day (day), fuel moisture code (ffmc, dmc, dc), initial spread index (content), weather conditions (temp, rh, wind, rain), and burned area (area). The first step in preprocessing is handling missing values. In this dataset, no missing values are found so that no special steps are needed to handle missing values. The next step is to convert categorical data into numeric format. Categorical data such as month and day are converted into numbers using the Label Encoding technique, as the XGBoost algorithm requires numerical input to perform predictive calculations.

After that, normalization of the numerical features is performed to ensure that all features are on the same scale. Normalization is important for the model to perform at its best. The normalized features include geographic location, fuel moisture code, initial spread index, air temperature, relative humidity, wind speed, rainfall, and burned area. After preprocessing is complete, the data is then divided into training set and testing set with a ratio of 80:20. This division aims to ensure fair model evaluation and prevent overfitting, where the model overfits the training data but undergeneralizes to new data.

The choice of data sharing ratio between training set and testing set is very important in machine learning as it affects how the model is trained and evaluated. A ratio of 80:20 is often chosen because it provides an optimal balance between the amount of data used to train the model and the amount of data used to evaluate the model. By using 80% of the data for training, the model has enough data to learn the patterns present in the dataset, while 20% of the data for testing is enough to give an accurate picture of the model's performance on data that it has never seen before. This ratio helps ensure that the model can generalize well from training data to unseen data, which is one of the main goals in machine learning.

The performance of the XGBoost model was evaluated using Mean Squared Error (MSE) and R-squared (R^2) metrics on the test dataset. To assess the impact of hyperparameter tuning, we compared the results of the XGBoost model with default parameters against the XGBoost model optimized through RandomizedSearchCV. The comparison of these results is summarized in the Table 2 :

Table 2. Result Comparison

Model	Test MSE	Test (R^2)
XGBoost (default parameters)	0.0216	0.9776
XGBoost (RandomizedSearchCV tuned)	0.0210	0.9820

Source: (Research Results, 2024)

The Table 2 illustrates that the XGBoost model with default parameters achieved a Test MSE of 0.0216 and an (R^2)value of 0.9776, indicating a high level of accuracy in predicting the target values. However, the XGBoost model optimized through RandomizedSearchCV showed a slight increase in Test MSE to 0.0210 and a corresponding decrease in (R^2)to 0.9820. Despite the slight increase in MSE and decrease in (R^2),the RandomizedSearchCV-tuned model demonstrates robust performance, reflecting the benefits of hyperparameter tuning in refining model parameters. The small differences in metrics suggest that both models are highly effective in making accurate predictions, but the optimized model may offer better generalization in diverse scenarios due to its fine-tuned parameters.

The Table 3 summarizes the best hyperparameters obtained from the RandomizedSearchCV optimization process for the XGBoost model.

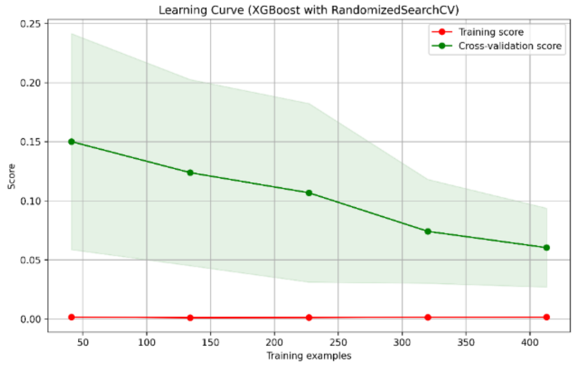
Table 3. Result Comparison

Hyperparameter	Value
colsample_bytree	0.7542
gamma	0.00798
learning_rate	0.0793
max_depth	6
n_estimators	466
subsample	0.8733

Source: (Research Results, 2024)

The colsample_bytree parameter, set at 0.7542, controls the fraction of features to be randomly sampled for each tree. A value of 0.7542 means that approximately 75.42% of the features are used to build each tree, helping to reduce overfitting by introducing feature randomness. The gamma parameter, with a value of 0.00798, determines the minimum amount of loss reduction needed to split a leaf node further within the tree. This value helps in controlling the complexity of the model by pruning branches with insufficient loss reduction. The learning_rate, set at 0.0793, scales the contribution of each tree by this factor, balancing the speed of learning and preventing overfitting. A max_depth of 6 indicates the maximum depth of a tree, controlling the complexity and the number of nodes in the trees. The n_estimators parameter, set to 466, specifies the number of trees to be built. The subsample parameter, with a value of 0.8733, determines the fraction of samples to be used for fitting each individual tree. This setting helps to prevent overfitting by adding randomness to the training process.

The learning curve for the XGBoost model with hyperparameter tuning using RandomizedSearchCV provides valuable insight into the performance and generalization ability of the model. As shown in Figure 2.



Source: (Research Results, 2024)
Figure 2. The learning curve

The red line represents the training score, which remains consistently low across different training sizes. This indicates that the model fits well on the training data with minimal error. A low training score indicates that the model does not suffer from high bias and can accurately capture patterns in the training data. The green line depicts the cross-validation score, which similarly decreases with an increase in the number of training examples. This trend indicates that the model generalizes well on unseen data, and its performance improves with more training data. The decrease in cross-validation error indicates that the model benefits from the additional data and can utilize more data to make more accurate predictions. The shaded region around the cross-validation score represents the variability or standard deviation of the cross-validation score across different folds. Narrower regions indicate more consistent performance across different subsets of data, while wider regions indicate greater variability. The shaded region narrows slightly as the number of training examples increases, indicating increased stability in model performance with more data.

PILAR Nusa Mandiri: Journal of Computing and Information System
Vol. 20, No. 2 September 2024 | DOI: 10.33480/pilar.v20i2.5569

These results highlight the importance of hyperparameter tuning in machine learning models. While default parameters can yield high performance, tuning parameters using RandomizedSearchCV can further refine the model, ensuring its robustness and reliability in various predictive tasks. The tuned model's performance underscores its potential for practical applications in predicting forest fire drought conditions, providing a valuable tool for forest management and risk assessment.

Discussion

Despite the slight increase in MSE and decrease in(R^2), the RandomizedSearchCV-tuned model demonstrates robust performance, reflecting the benefits of hyperparameter tuning in refining model parameters. The small differences in metrics suggest that both models are highly effective in making accurate predictions, but the optimized model may offer better generalization in diverse scenarios due to its fine-tuned parameters. Comparing these results with previous studies, it is evident that hyperparameter tuning plays a crucial role in enhancing model performance. For instance, (Ali et al., 2023) highlighted the efficiency of RandomizedSearch over GridSearch in hyperparameter optimization, demonstrating improved performance with less computational cost. Similarly, (Ehsani & Hosseini, 2024) used RandomizedSearchCV to optimize XGBoost for retail sales prediction and reported significant improvements in accuracy compared to using default parameters.

(Sun et al., 2024) found that XGBoost outperformed other algorithms like Random Forest and SVM in predicting forest fire occurrences, underscoring the strength of XGBoost in handling complex predictive tasks. Our findings align with these studies, reinforcing the value of hyperparameter tuning in extracting the best performance from machine learning models. The tuned model's performance underscores its potential for practical applications in predicting forest fire drought conditions, providing a valuable tool for forest management and risk assessment. By improving the accuracy of predictions, forest managers can better prepare for and mitigate the impacts of forest fires, ultimately contributing to more effective environmental and resource management.

These results highlight the importance of hyperparameter tuning in machine learning models. While default parameters can yield high performance, tuning parameters using RandomizedSearchCV can further refine the model, ensuring its robustness and reliability in various predictive tasks.

CONCLUSION

This study aimed to enhance the predictive accuracy of forest fire drought conditions using the XGBoost algorithm, optimized through RandomizedSearchCV. The findings reveal that while the XGBoost model with default parameters exhibited high performance with a Test MSE of 0.0216 and an (R^2)value of 0.9776, the model optimized through hyperparameter tuning with RandomizedSearchCV achieved a Test MSE of 0.0210 and an (R^2)value of 0.9820. These results indicate that both models are effective in making accurate predictions, with the tuned model demonstrating robust performance and better generalization capabilities.

The slight differences in performance metrics underscore the importance of hyperparameter optimization in refining model accuracy and reliability. The comparison with previous studies further supports the value of RandomizedSearchCV in efficiently enhancing machine learning models. This optimized approach is particularly beneficial in practical applications, where accurate and reliable predictions of forest fire drought conditions are crucial for effective forest management and risk assessment.

Overall, the study confirms that hyperparameter tuning, while computationally intensive, significantly contributes to the development of high-performance predictive models. The insights gained from this research provide a valuable foundation for future work aimed at improving prediction systems for forest fire management, thereby aiding in the development of more resilient and proactive environmental strategies.

REFERENCE

Alamsyah, N., Budiman, Danestiara, V. R., Akbar, I., & Setiana, E. (2023). Optimizing Computational Efficiency in Feature Selection for Machine Learning Models: A Study Crime Detection Based on Criminal Data. *2023 Eighth International Conference on Informatics and Computing (ICIC)*, 1–6. https://doi.org/10.1109/ICIC60109.2023.10382057

Alamsyah, N., Saparudin, & Kurniati, A. P. (2023). A Novel Airfare Dataset To Predict Travel Agent Profits Based On Dynamic Pricing. *2023 11th International Conference on Information and Communication Technology (ICoICT)*, 575–581. https://doi.org/10.1109/ICoICT58202.2023.10262694

Alamsyah, N., Yoga, T. P., & Budiman, B. (2024). IMPROVING TRAFFIC DENSITY PREDICTION USING LSTM WITH PARAMETRIC ReLU (PReLU) ACTIVATION. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *9*(2), 154–160. https://doi.org/10.33480/jitk.v9i2.5046

Albahra, S., Gorbett, T., Robertson, S., D'Aleo, G., Kumar, S. V. S., Ockunzzi, S., Lallo, D., Hu, B., & Rashidi, H. H. (2023). Artificial intelligence and machine learning overview in pathology & laboratory medicine: A general review of data preprocessing and basic supervised concepts. *Seminars in Diagnostic Pathology*, *40*(2), 71–87. https://doi.org/10.1053/j.semdp.2023.02.002

Ali, Y. A., Awwad, E. M., Al-Razgan, M., & Maarouf, A. (2023). Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity Yasser. *Processes*, *11*(2), 1–21. https://doi.org/10.3390/pr11020349

Amal Asselman, M. K., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments*, *31*(6), 3360–3379. https://doi.org/10.1080/10494820.2021.1928235

Baseer, K. K., Nas, S. B. A., Dharani, S., Sravani, S., Yashwanth, P., & Jyothirmai, P. (2023). Medical Diagnosis of Human Heart Diseases with and without Hyperparameter tuning through Machine Learning. *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, 1–8. https://doi.org/10.1109/ICCMC56507.2023.10084156

Carta, F., Zidda, C., Putzu, M., Loru, D., Anedda, M., & Giusto, D. (2023). Advancements in Forest Fire Prevention: A Comprehensive Survey. *Sensors*, *23*(14), 1–26. https://doi.org/10.3390/s23146635

Ehsani, F., & Hosseini, M. (2024). Customer purchase prediction in electronic markets from clickstream data using the Oracle meta-classifier. *Operational Research*, *24*(1), 11. https://doi.org/10.1007/s12351-023-00813-6

Erpurini, W., Putrada, A. G., Alamsyah, N., Pane, S. F., & Nurkamal Fauzan, M. (2023). Confirmatory Factor Analysis for The Impact of Students' Social Medial on University Digital Marketing. *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 615–620. https://doi.org/10.1109/ICCoSITE57641.2023.10127744

Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., & Shao, L. (2023). Normalization Techniques in Training DNNs: Methodology, Analysis and Application. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *45*(08), 10173–10196. https://doi.org/10.1109/TPAMI.2023.3250241

Jones, M. W., Abatzoglou, J. T., Veraverbeke, S., Andela, N., Lasslop, G., Forkel, M., Smith, A. J. P., Burton, C., Betts, R. A., van der Werf, G. R., Sitch, S., Canadell, J. G., Santín, C., Kolden, C., Doerr, S. H., & Le Quéré, C. (2022). Global and Regional Trends and Drivers of Fire Under Climate Change. *Reviews of Geophysics*, *60*(3), e2020RG000726. https://doi.org/10.1029/2020RG000726

Korená Hillayová, M., Holécy, J., Korísteková, K., Bakšová, M., Ostrihoň, M., & Škvarenina, J. (2023). Ongoing climatic change increases the risk of wildfires. Case study: Carpathian spruce forests. *Journal of Environmental Management*, *337*, 117620. https://doi.org/10.1016/j.jenvman.2023.117620

Putrada, A. G., Alamsyah, N., Pane, S. F., & Fauzan, M. N. (2022). XGBoost for IDS on WSN Cyber Attacks with Imbalanced Data. *2022 International Symposium on Electronics and Smart Devices (ISESD)*, 1–7. https://doi.org/10.1109/ISESD56103.2022.9980630

Qin, Y., Ju, W., Wu, H., Luo, X., & Zhang, M. (2024). Learning Graph ODE for Continuous-Time Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *36*(7), 3224–3236. https://doi.org/10.1109/TKDE.2024.3349397

Sun, X., Li, N., Chen, D., Chen, G., Sun, C., Shi, M., Gao, X., Wang, K., & Hezam, I. M. (2024). A Forest Fire Prediction Model Based on Cellular Automata and Machine Learning. *IEEE Access*, *12*, 55389–55403. https://doi.org/10.1109/ACCESS.2024.3389035

THYGE PEDERSEN, U. (2023). *Forest Fires* [dataset]. https://www.kaggle.com/datasets/ulrikthygepedersen/forest-fires

Todorovic, M., Stanisic, N., Zivkovic, M., Bacanin, N., Simic, V., & Tirkolaee, E. B. (2023). Improving audit opinion prediction accuracy using metaheuristics-tuned XGBoost algorithm with interpretable results through SHAP value analysis. *Applied Soft Computing*, *149*, 110955.

https://doi.org/10.1016/j.asoc.2023.110955

Wu, L., Li, X., Yuan, J., & Wang, S. (2023). Real-time prediction of tunnel face conditions using XGBoost Random Forest algorithm. *Frontiers of Structural and Civil Engineering*, *17*(12), 1777–1795. https://doi.org/10.1007/s11709-023-0044-4

Zhao, Y., Zhang, W., & Liu, X. (2024). Grid search with a weighted error function: Hyper-parameter optimization for financial time series forecasting. *Applied Soft Computing*, *154*, 111362. https://doi.org/10.1016/j.asoc.2024.111362