J-Icon : Jurnal Informatika dan Komputer
Vol. 13 No. 2, October 2025, pp. 77~84
DOI: 10.35508/jicon.v13i2.18493

Hyperparameter Optimization in Machine Learning Models on Sky Survey Data Classification

Efraim Kurniawan Dairo Kette*

Fakultas Sains dan Teknik, Universitas Nusa Cendana, Kupang, Indonesia
Email*: efraim.kette@staf.undana.ac.id

ABSTRACT

Discovering the optimal model in today's popularity of various machine learning applications remains an essential challenge. Besides data dependency, the performance of classification models is also affected by deciding on suitable algorithm with optimal hyperparameter settings. This study conducted a hyperparameter optimization process and compared the accuracy results by applying various classification models to the observation dataset. This study obtains data from the Sloan Digital Sky Survey Data Release 18 (SDSS-DR18) and Sloan Extension for Galactic Understanding and Exploration (SEGUE-IV). The SDSS-DR18 and SEGUE-IV provide observational data of space objects, such as stellar spectra with corresponding positions and magnitudes of galaxies or stars. The SDSS-DR18 dataset contains magnitude and redshift data of celestial objects with target features of stars, Quasi Stellar Objects (QSOs), and galaxies. The SEGUE-IV dataset contains equivalent-width parameters, inline indices, and other features to the radial velocity of the corresponding star spectrum. This study utilized several machine learning models, such as k-Nearest Neighbor (KNN), Gaussian-Naive Bayes, eXtreme Gradient Boosting (XGBoost), Random Forest, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP). This study utilized Bayesian, Grid, and Random-based approaches to find the optimal hyperparameters to maximize the performance of the classification model. This study proved that some classification models have improved accuracy scores through the Bayesian-based hyperparameter optimization settings. This study discovers the XGBoost model shows the highest classification results after hyperparameters optimization compared to other models for both datasets with an average accuracy of 99.10% and 95.11%, respectively.

Keywords: Machine Learning, Hyperparameter Optimization, Sky Object Classification.

1. INTRODUCTION

Classification with machine learning has become a powerful tool for various applications [1]. Machine learning capabilities allow computers to perform classification processes based on patterns and characteristics learned in the data. Despite the rising popularity of machine learning applications, identifying which classification models consistently produce the most accurate results remains a crucial challenge. The quality, amount, and relevance of the data utilized in the training process have a significant impact on the classification model's performance [2]. However, beyond the data, the efficiency of classification models is also determined by selecting the best algorithms with optimal hyperparameter settings. Therefore, a systematic approach is essential to identify the best-suited algorithms with the most suitable model settings for achieving optimal classification performance.

Machine learning algorithms are often used in science, especially astronomy. Over the past few years, the Sloan Digital Sky Survey (SDSS) project, which attempts to map a quarter of the sky, has produced much observational data [3]. SDSS provides observational data in the form of physical, atmospheric, and spectral parameters of various celestial objects that are freely accessible. Astronomers were gradually unable to manually categorize and label celestial objects in the future due to the vast amount of data with which newly discovered celestial objects. Thus, utilizing machine learning classification algorithms is meant to help overcome this problem.

Several categorization algorithms efficiently process vast amounts of data already developed into classification models. It is used to predict a target variable from new data. Classification models with diverse methodologies are often used for sky object classification problems [4][5]. In this study, several classification models are applied, such as k-nearest Neighbor (KNN), Gaussian Naive Bayes, Support Vector Machines (SVM), Random Forest, eXtreme Gradient Boosting (XGBoost), and Multi-Layer Perceptron (MLP), to classify sky object observation data. The models used in this study were selected

*) **Corresponding Author**
Submitted : September 19, 2024
Accepted : August 15, 2025
Published : September 26, 2025

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

based on their simplicity and ease of interpretation (Random Forest, Gaussian Naive Bayes, and KNN), as well as their ability to capture the underlying patterns of the data and produce high accuracy (XGBoost and MLP)[6].The performance of each classification model is shown by comparing the categorized results. The classification results from the various algorithms will be compared to determine the performance of each classification model created.

This study aims to obtain optimal performances through configuration variable settings known as hyperparameters. They differ from parameters, which are variables derived from data. Machine learning model hyperparameters are tuned manually before the training process [7]. It also regulates the algorithm's structure and complexity. Grid-based and randomized searches are the most common approaches in determining optimal hyperparameters [8]. Grid-based search systematically uses all possible hyperparameter settings, requiring high computational time. Random-based search uses a subset of samples taken from the overall hyperparameter settings, showing efficiency but not comprehensiveness. Bayesian-based hyperparameter search applies a probabilistic approach to all possible hyperparameter settings, thus showing consistency and faster convergence to the optimal hyperparameter settings [9]. Therefore, Bayesian-based search is applied in this study to find the optimal hyperparameters for each model.

2. MATERY AND METHODOLOGY

Data

This study uses observational data from the Sloan Digital Sky Survey 18th release (SDSS-DR18) and Sloan Extension for Galactic Understanding and Exploration (SEGUE-IV). The SDSS dataset contains magnitude and redshift data of celestial objects with target variables of stars, Quasi Stellar Objects (QSOs), and galaxies. This study obtained 10,000 data samples from the SDSS-DR18 source and divided them into three classes. The obtained data consist of 4795, 4089, and 1116 samples for the galaxy, star, and the QSO class, respectively. Each sample has eighteen features such as *objid, ra, dec, u, g, r, i, z, run, rerun, camcol, field, specobjid, redshift, plate, mjd, fiberid*, and *class* as classification target.

The SEGUE-IV dataset included in SDSS-V contains an equivalent width of several stellar spectral lines [10]. This study obtained 4148 data samples from the SDSS-V Stellar Parameter Pipeline (*sppLines*) table. Each sample in the SEGUE dataset has 78 parameters, with the *teffadopt* feature as the classification target, which is the average stellar effective temperature calculated in various ways. As shown in Table 1, each sample *teffadopt* feature is divided into specific spectral classes for the target classification model.

Table 1. Spectral Classes of *teffadopt*

Spectral Classes (label)	***teffadopt* (°K)**
O	28,000 - 50,000
B	10,000 - 28,000
A	7,500 - 10,000
F	6,000 - 7,500
G	4,900 - 6,000
K	3,500 - 4,900
M	2,000 - 3,500
L	< 2,000

Preprocessing

This study removed six features, such as *objid*, *run*, *rerun*, *camcol*, and *field*, from the first dataset (SDSS DR-18 dataset) that were unrelated to the classification process. The features *u*, *g*, *r*, *i*, *z* (*better of*DeV/Exp *magnitude fit*) are Thuan Gunn astronomical magnitude systems representing the response of the 5-band telescope. This study finds a high correlation between these several features. These features are then simplified by the Principal Component Analysis (PCA) method into three new features called *PCA*_1, *PCA*_2, and *PCA*_3 to accelerate the convergence of the classification process as done in previous studies [11]. Reducing features were not performed on the second dataset (SEGUE-IV dataset). The correlation of each feature was low in the second dataset. As a result, all features from the second dataset were retained so models could learn more information. This study utilizes the minimum-maximum scaling method (*MinMaxScaler*) with a minimum value limit of 0 and 1 for the maximum limit to reduce the impact of the different values in the two datasets. Any missing value samples in both datasets will be removed from the overall dataset if more than 20% of the total data. Meanwhile, the median of the valid values will be given as an estimation of the samples if the missing values do not exceed 20% of the total data.

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

Model

This research uses several classification models, such as k-Nearest Neighbor (KNN), Gaussian-Naive Bayes, eXtreme Gradient Boosting (XGBoost), Random Forest, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP).

1. Gaussian-Naive Bayes

Naive Bayes is a parametric or non-parametric classification algorithm based on Bayesian concepts [12]. This method is usually more appropriate for categorical datasets [13]. This study utilizes the Gaussian (normalized) distribution function to improve the performance of the Naive Bayes method in handling continuous data. There is the equation (1), (2), and (3).

$$\mu = \frac{\sum_{j=1}^{n} v_j}{n} \quad (1)$$

$$\sigma^2 = \sqrt{\frac{\sum_{j=1}^{n}(v_j - \mu_k)^2}{n-1}} \quad (2)$$

$$P(E_i = v_j | C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}} \quad (3)$$

Equation (3) calculated the probability of a possible classification [14]. $P(E_i = v|C_k)$ denoted the probability of an attribute/feature in a particular class, where the C_k $\{C_1, C_2, \dots, C_l\}$ denoted the *k*–th known label or class, and $E_i\{E_1, E_2, \dots, E_m\}$ denoted the *i*–th attribute or features. Based on equation (1) and (2), μ_k and σ_k^2 denoted the mean and standard deviation parameters, respectively. v_j $\{v_1, v_2, \dots, v_n\}$ denotes *j*–th value of a particular attribute/feature. *m* and *n* as the number of attributes/features and data, respectively. Lastly, *l* denotes the number of labels or data classes.

2. k-Nearest Neighbors (KNN)

KNN is simple non-parametric classification algorithm that categorizes a vector of new sample data (y; $y \in \mathbb{R}^m$) against most training data with a class determined by the nearest *k*-value [15]. The nearest neighbor of *y* is the closest distance of the new sample to each training data sample (x_i; $1 \leq i \geq n$ where *n* is the number of data).

$$d(y, x_i) = \| y - x_i \|_2 \quad (4)$$

$$l_y = \arg\max_c \left(\sum_{(x_i^{NN}, l_i^{NN}) \in T_k} \delta(c = l_i^{NN}) \right) \quad (5)$$

In Equation (4), $d(y, x_i)$ denoted the distance function between the data. The l_2 norm is a form of Euclidean distance that determines the closest distance between the new sample data and the training data. In Equation (5), l_y denoted the prediction of the class label of sample data *y*, with $arg\ max$ as an argument that gives the highest value to the function, $\delta(c = l_i^{NN})$ denoted the Dirac-delta function (1 if met the condition and 0 otherwise).

3. Support Vector Machine

SVM is a classification algorithm that uses the concept of hyperplanes that separate classes in feature space [16]. SVM is effective for handling high-dimensional issues with limited training data [17].

$$f(x) = w^T x_i + b_i \quad (6)$$

In Equation (6), $f(x)$ denoted decision function that calculates the distance from the data point to the hyperplane boundary region. The $f(x)$ is positive if the data point is classified into the positive class $\{w^T x_i + b_i \geq 1\}$, and negative if the data point is classified into the negative class $\{w^T x_i + b_i < 1\}$. w^T denoted the transpose of the weight vector perpendicular to the hyperplane that determines the orientation of the hyperplane. b_i denoted the intercept line parallel to the hyperplane boundary line.

4. Random Forest and eXtreme Gradient Boosting (XGBoost)

Random Forest and XGBoost are ensemble classification algorithms that utilize the concept of decision trees to make predictions [18] [19]. The decision tree model was developed independently in the Random Forest model, and the final forecast is usually made by classification of the individual tree projections. The decision tree in the XGBoost model is built sequentially with a gradient descent-based optimization process.

$$f_i(x) = \sum_{k=1}^{K} w_{k,i} \cdot I(x \in R_{k,i}) \quad (7)$$

In Equation (7), $f_i(x)$ denoted the predicted probability or class label of the *i*-th tree for data point *x*. The notation *K* is the number of leaves in the decision tree, with $w_{k,i}$ denoted the weight of leaf-*k* of

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

tree-i. $I(x \in R_{k,i})$ denotes an indicator function, which takes the value one if x as the input belongs to leaf-k of tree-i and zero otherwise.

5. Multi-Layer Perceptron (MLP)
MLP is a classification algorithm that often uses the back-propagation method, which is usually combined with gradient descent to adjust the weights and bias to minimize the loss function [20].

$$z_i = W_{i,k} x_i + b \quad (8)$$

In Equation (8), z_i denoted the i-th hidden layer that connects the input to the output layer. The x_i denoted the i-th input data vector and the $W_{i,k}$ denoted the weight matrix from the i-th input vector to the kth node of the i-th hidden layer.

Evaluation methods and metrics

This study utilized cross-validation as a method to evaluate the model's performance. Cross-validation provides a better approach to distributing training and testing data and is fair in distributing data evenly for the evaluation process [21]. This study utilized the accuracy score as a metric to measure the cross-testing process of each classification model. The accuracy score is calculated to show the ratio of correct classification results to total data. This study used ten cross-tests to evaluate each classification procedure, producing an average accuracy across all tests.

3. RESULT AND DISCUSSIONS

This study conducts modeling experiments and compares the classification results of optimized models on a personal computer with AMD Ryzen 5 5500U CPU specifications (2.1 GHz), 16 GB of memory, and Microsoft 11 operating system. This study utilizes the Python programming language with the Jupyter Notebook compiler. Several tools used to help build the model were obtained from TensorFlow version 2.9.1. Tables 2 and 3 show comparisons of classification experiments results towards sky object data consisting of stars, galaxies, and QSOs from the SDSS-DR18 observational dataset. Table 2 shows the results of each model using hyperparameter settings set based on grid-based (KNN, Gaussian-Naive Bayes, and SVM) and random-based (Random Forest, XGBoost, and MLP) searches.

Table 2. Classification of sky objects (stars, galaxies, and QSOs) with grid-based and random-based hyperparameter optimization.

Model	Avg. Accuracy (%)	Optimization time (seconds)	Training Duration (seconds)
MLP	98.73	372.617871	678.905342
Random Forest	98.64	17.399989	31.267240
XGBoost	98.36	8.486558	16.335466
Gaussian-Naïve Bayes	98.03	2.884586	0.037334
SVM	96.95	9.154659	9.245392
KNN	91.54	30.870617	0.300714

Based on the results and computation time from Table 2, the MLP model shows the highest level of accuracy compared to other models for the classification of celestial objects (stars, galaxies, and QSOs) with an average cross-test score of 98.73%. Meanwhile, the KNN model shows the lowest accuracy with an average cross-test score of 91.54% for the same data classification. Table 3 shows comparisons of classification experiments result towards SDSS-DR18 dataset but utilize the hyperparameter settings set based on Bayesian search. Table 3 shows that few model accuracies increased, although there is no significant difference compared to the results obtained in Table 2.

Table 3. Classification of sky objects (Stars, Galaxies, and QSOs) with Bayesian-based hyperparameter optimization.

Model	Avg. Accuracy (%)	Optimization time (seconds)	Training Duration (seconds)
XGBoost	99.10	361.407815	7.702596
MLP	98.73	284.410771	16.377885
Random Forest	98.68	543.397319	14.029526
Gaussian-Naïve Bayes	98.03	34.873845	0.038479
SVM	96.95	247.178102	9.036567
KNN	91.87	57.822821	0.330243

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

The XGBoost model shows the highest average score of 99.10%. Meanwhile, the KNN model still shows the lowest average score of 91.87%. The optimal hyperparameter settings of the Bayesian-based search results for each model against the SDSS-DR18 dataset are shown in Table 4.

Table 4. Optimal hyperparameter settings for sky object classification (Stars, Galaxies, and QSOs)

Model	***hyperparameter* settings**
XGBoost	'*colsample_bytree*': 0.7, '*gamma*' : 0.3, '*learning_rate*' : 0.05, '*max_depth*' : 4, '*min_child_weight*': 3
MLP	'*solver*' : 'lbfgs', '*max_iter*': 10000, '*learning_rate_init*': 0.01, '*hidden_layer_sizes*': (32, 64), 'alpha' : 0.01, '*activation*' : 'relu'
Random Forest	'*max_depth*' : 80, '*max_features*' : 3, '*min_samples_leaf*': 3, '*min_samples_split*': 8
Gaussian-Naïve Bayes	'*var_smoothing*' : 6.579332246575682e-08
SVM	'*C*' : 30, '*kernel*': 'rbf'
KNN	'*metric*' : 'manhattan', '*n_neighbors*': 4, '*weights*' : '*distance*'

Table 5 and 6 show comparisons of classification experiments results towards stellar (Stars) spectrum data (SEGUE-IV). Table 5 shows the results of each model using hyperparameter settings set based on grid-based (KNN, Gaussian-Naive Bayes, and SVM) and random-based (Random Forest, XGBoost, and MLP) searches. Table 5 results show that the XGBoost model has the highest accuracy compared to other models on classification stellar (stars) spectrum class with an average score of 94.89%. Meanwhile, the Gaussian-Naïve Bayes model shows the lowest accuracy with an average score of 88.67%. Table 6 shows comparisons of classification experiments result towards SEGUE-IV dataset but utilize the hyperparameter settings set based on Bayesian search.

Table 5. Classification of stellar (stars) spectrum class with grid-based and random-based hyperparameter optimization.

Model	Avg. Accuracy (%)	Optimization time (seconds)	Training Duration (seconds)
XGBoost	94.8891	22.993319	9.086410
SVM	94.8411	1.806240	0.950489
Random Forest	94.6966	17.532088	66.116585
MLP	94.3350	63.287882	21.337056
KNN	93.8284	19.782861	0.761138
Gaussian-Naïve Bayes	88.6698	4.044270	0.072424

Table 5 results show that the XGBoost model has the highest accuracy compared to other models on classification stellar (stars) spectrum class with an average score of 94.89%. Meanwhile, the Gaussian-Naïve Bayes model shows the lowest accuracy with an average score of 88.67%. Table 6 shows comparisons of classification experiment results with the SEGUE-IV dataset, but it utilizes the hyperparameter settings set based on Bayesian search. After conducting hyperparameter optimization experiments using grid-based, random, and Bayesian approaches, MLP, SVM, and Gaussian Naive Bayes (GNB) showed minimal accuracy improvements due to inherent model limitations and data compatibility issues. GNB's limitation of feature independence made it fundamentally incompatible with the second dataset, as real-world data rarely meets this assumption. SVM's performance is heavily dependent on kernel choice. Experimenting with the second dataset with 78 features, the kernel likely already operated near-optimally, leaving little room for improvement through tuning parameters like C or gamma. Similarly, MLP flexibility was constrained by the smaller size (4,148 samples) of the second dataset, limiting its ability to benefit from deeper architectural tuning. Therefore, we conclude that these models' performance was bottlenecked by their design and data compatibility, making hyperparameter optimization less impactful compared with models like XGBoost and Random Forest, which are inherently more adaptable to the datasets structures.

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

Table 6. Classification of stellar (stars) spectrum class with Bayesian-based hyperparameter optimization.

Model	Avg. Accuracy (%)	Optimization time (seconds)	Training Duration (seconds)
XGBoost	95.1060	982.638881	14.582254
SVM	94.8411	79.392844	0.940902
Random Forest	94.7449	491.712202	6.966766
MLP	93.9005	235.157569	31.069682
KNN	93.8042	77.947090	0.748705
Gaussian-Naïve Bayes	88.6698	25.142494	0.069767

Table 7. Optimal hyperparameter settings for stellar (stars) spectrum class.

Model	***hyperparameter* settings**
XGBoost	'*colsample_bytree'*: 0.4, '*gamma'* : 0.4, '*learning_rate'* : 0.2, '*max_depth'* : 10, '*min_child_weight'*: 1
SVM	'*C*' : 10, '*kernel*': 'linear'
Random Forest	'*max_depth'* : 80, '*max_features'* : 3, '*min_samples_leaf*': 3, '*min_samples_split'*: 12
MLP	'*solver'* : 'adam, '*max_iter'*: 10000, '*learning_rate_init'*: 0.01, '*hidden_layer_sizes'*: (8, 16), 'alpha' : 0.01, '*activation'* : 'relu'
KNN	'*metric*' : 'manhattan', '*n_neighbors*': 10, '*weights*' : '*distance*'
Gaussian-Naïve Bayes	'*var_smoothing'* : 0.3511191734215131

4. SUMMARY

This study utilized six supervised learning algorithms to classify two observational datasets obtained from the Sloan Digital Sky Survey 18th release (SDSS-DR18) and Sloan Extension for Galactic Understanding and Exploration (SEGUE-IV). The SDSS-DR18 dataset contains 10000 samples with features such as magnitude and redshift of stellar objects, and target features consist of Stars, QSOs, and Galaxies. The SEGUE-IV dataset contains 4148 samples with features such as equivalent-width parameters, inline indices, and other radial velocity features to the corresponding star spectrum. This study utilizes a Bayesian-based hyperparameter search to determine the optimal settings that maximize the performance of the classification model.

This study shows that the XGBoost algorithm provides the best performance compared to the other models for both datasets, with an average accuracy of 99.10% and 95.11%, respectively. Despite resource constraints using personal computers for this study, the experiment successfully achieved high accuracy goals for large-scale data. It proves the model's effectiveness for large-scale sky survey classification. While optimization time for the XGBoost model took forty-two times longer, it is a notable trade-off because accuracy is often mission-critical in scientific applications. This study found that the optimal hyperparameter settings of the XGBoost model for classification of the SDSS-DR18 database include *colsample_*bytree of 0.7, *gamma* of 0.3, *learning_rate* of 0.05, *max_depth* of 4, and *min_child_weight* of 3. Meanwhile, the optimal hyperparameter settings for the XGBoost model for classification of the SEGUE-IV database include *colsample_bytree* of 0.4, *gamma* of 0.4, *learning_rate* of 0.2, *max_depth* of 10, and *min_child_weight* of 1.

This study proved that some classification models have improved accuracy scores through the Bayesian-based hyperparameter optimization. We intentionally avoided handling data imbalance to maintain the robustness and real-world applicability of the models. By keeping the datasets in their original imbalanced state, we ensured that the results reflected how the models would perform on primary data, which is critical for evaluating their suitability for real-world sky object classification tasks. Introducing imbalance-handling techniques (e.g., over-sampling, under-sampling, or class weighting) could artificially inflate accuracy or mask weaknesses, leading to biased conclusions about model performance. However, future work could explore these techniques to address potential skews in class distributions, as imbalance

handling might improve minority class recall and overall generalizability, especially for models like GNB or KNN that struggle with imbalanced data. Also, this could provide a more comprehensive understanding of the model's behavior across different data scenarios. Further improvement can explore the possibilities of other ensemble classification techniques that may outperform the XGBoost algorithm.

REFERENCES

[1] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, pp. 160, 2021, doi: 10.1007/s42979-021-00592-x.
[2] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 6, pp. 420, 2021, doi: 10.1007/s42979-021-00815-1.
[3] A. Almeida *et al.*, "The Eighteenth Data Release of the Sloan Digital Sky Surveys: Targeting and First Spectra from SDSS-V," *Astrophys. J. Suppl. Ser.*, vol. 267, no. 2, pp. 44, 2023, doi: 10.3847/1538-4365/acda98.
[4] S. Ethiraj and B. K. Bolla, "Classification of Quasars, Galaxies, and Stars in the Mapping of the Universe Multi-modal Deep Learning," *Deep Learning Developers Conference*, 2022, doi: 10.48550/arXiv.2205.10745.
[5] W. Xiao-Qing and Y. Jin-Meng, "Classification of star/galaxy/QSO and Star Spectral Types from LAMOST Data Release 5 with Machine Learning Approaches," *Chin. J. Phys.*, vol. 69, pp. 303–311, 2021, doi: 10.1016/j.cjph.2020.03.008.
[6] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," 2020, doi: 10.48550/arXiv.1811.12808.
[7] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.
[8] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012. [Online]. Available at: https://dl.acm.org/doi/pdf/10.5555/2188385.2188395.
[9] R. Turner *dkk.*, "Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020," 2021, doi: 10.48550/arXiv.2104.10201.
[10] J. Kollmeier *dkk.*, "SDSS-V Pioneering Panoptic Spectroscopy," vol. 51, pp. 274, 2019. [Online]. Available at: https://arxiv.org/pdf/1711.03234.
[11] F. Oktariani, F. A. Adziima, and E. K. D. Kette, "Application of Classification Algorithms in Machine Learning on Digital Sloan Data Survey (SDSS) and Sloan Extension for Galactic Understanding and Exploration (SEGUE) data," *13th International Symposium on Computational Science*, 2022. [Online]. Available at: https://iscs2022.site/event/1.
[12] M. Ben-Bassat, K. L. Klove, and M. H. Weil, "Sensitivity Analysis in Bayesian Classification Models: Multiplicative Deviations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 3, pp. 261–266, Mei 1980, doi: 10.1109/TPAMI.1980.4767015.
[13] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 1995, pp. 338–345.
[14] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Mach. Learn.*, vol. 29, no. 2, pp. 103–130, 1997, doi: 10.1023/A:1007413511361.
[15] P. K. Syriopoulos, N. G. Kalampalikis, S. B. Kotsiantis, and M. N. Vrahatis, "kNN Classification: a review," *Ann. Math. Artif. Intell.*, 2023, doi: 10.1007/s10472-023-09882-x.
[16] T. Evgeniou and M. Pontil, "Support Vector Machines: Theory and Applications," *Machine Learning and Its Applications: Advanced Lectures*, 2001, pp. 249–257. doi: 10.1007/3-540-44673-7_12.
[17] A. Roy and S. Chakraborty, "Support Vector Machine in Structural Reliability Analysis: A Review," *Reliab. Eng. Syst. Saf.*, vol. 233, pp. 109126, 2023, doi: 10.1016/j.ress.2023.109126.
[18] P. Dutta, S. Paul, and A. Kumar, "Chapter 25 - Comparative Analysis of Various Supervised Machine Learning Techniques for Diagnosis of COVID-19," dalam *Electronic Devices, Circuits, and Systems for Biomedical Applications*, 2021, pp. 521–540. doi: 10.1016/B978-0-323-85172-5.00020-4.
[19] I. A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, and A. El-Shafie, "Extreme Gradient Boosting (XGBoost) Model to Predict the Groundwater Levels in Selangor Malaysia," *Ain Shams Eng. J.*, vol. 12, no. 2, pp. 1545–1556, 2021, doi: 10.1016/j.asej.2020.11.011.
[20] S. S. Chai, W. L. Cheah, K. L. Goh, Y. H. R. Chang, K. Y. Sim, and K. O. Chin, "A Multilayer Perceptron Neural Network Model to Classify Hypertension in Adolescents Using Anthropometric

Measurements: A Cross-Sectional Study in Sarawak, Malaysia," *Comput. Math. Methods Med.*, vol. 2021, no. 1, pp. 2794888, 2021, doi: 10.1155/2021/2794888.

[21] T.T. Wong, "Performance Evaluation of Classification Algorithms by *k*-Fold and Leave-One-Out Cross Validation," *Pattern Recognit.*, vol. 48, no. 9, pp. 2839–2846, 2015, doi: 10.1016/j.patcog.2015.03.009.

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)