



## **Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure**

**Dimas Aryo Anggoro<sup>1\*</sup>**

**Salsa Sasmita Mukti<sup>1</sup>**

<sup>1</sup>*Department of Informatics, Universitas Muhammadiyah Surakarta, Indonesia*

\* Corresponding author's Email: [dimas.a.anggoro@ums.ac.id](mailto:dimas.a.anggoro@ums.ac.id)

---

**Abstract:** The kidneys have an essential role in the body; if it does not work properly, it will cause disease, one of which is chronic kidney failure (CRF). Therefore, a machine learning algorithm is needed to help predict CRF, one of which is the Extreme Gradient Boosting (XGBoost) algorithm. However, XGBoost has one thing that must be considered, namely the presence of hyperparameters that need to be tuned. In this research, tuning was done using grid search and random search methods, which would be compared to get a good performance and help predict GGK. In the dataset used, several features were incomplete, so that data cleaning was carried out. Based on the unbalanced amount of data, oversampling with SMOTE and random oversampling methods were needed to balance the data. Next, the data were normalized using two: min-max and z-score normalization. After the data had been normalized, a splitting process was performed with a division of 70% for training data and 30% for testing data. Furthermore, the hyperparameter was tuned to obtain the optimal parameter value, which was then performed data processing and evaluated to obtain the accuracy and f-measure values of the tuned parameter values. In the grid search method with data performed with min-max and z-score normalization combined with oversampling random oversampling, the best results were obtained with an accuracy value of 99.28% and an f-measure value of 0.9942. Comparison with other methods was carried out and the proposed method obtained an accuracy of 99.33% and f-measure 1.0.

**Keywords:** XGBoost, Grid search, Random search, Chronic kidney failure.

---

### **1. Introduction**

The kidneys are one of the most important parts of the body. The kidneys have the role of filtering the blood and removing waste and toxins in the body. If an essential part of the body cannot work properly, it will cause a problem. Likewise, a kidney that cannot work properly will cause a disease. If left to linger, the disease will become more chronic, one of which is chronic kidney failure (CRF). CRF is a disease with abnormalities in the kidneys' function or/and structure that lasts for months, even years [1].

CRF is a disease of global concern. One of them, India, had about 229 patients with CRF per one million population, and there were more than 100,000 patients with CRF undergoing kidney replacement; the data have predicted that patients with CRF would increase [2]. Based on data sourced from Global

Burden of Disease in 2010, the global cause of death ratio of CRF was 27th, with 15.7 per 100,000 cases, and the ranking increased to 18th place, with a death ratio of 16.3 per 100,000 cases [3]. Other data disclosed by the United States Renal Data System in America stated that the prevalence of CRF increased by 20-25% each year [4]. Also, according to data compiled by the National Center for Health Statistics (NHANES), the prevalence of CRF in developed countries reached 12-20% [5].

In the last ten years, CRF is a disease that has become a major health problem [6]. Therefore, a machine learning algorithm is needed to detect disease. Machine learning has been widely used in the health world to detect various diseases and risk factors. Breast cancer prediction [7] and prediction of heart failure patient survival [8] are examples of machine learning applications in detecting disease in the world of health. Algorithms in machine learning

are expected to assist in the classification process accurately. One of the algorithms that can be used is Extreme Gradient Boosting (XGBoost).

XGBoost is a machine learning method with good scalability and high processing speed so that the results obtained can have a high value. Based on research on CRF conducted by Raju [9] using several algorithms and XGBoost got the best results. However, some things are less of a concern, such as the hyperparameters in XGBoost. Hyperparameter adjustment is required to attain optimal results using the XGBoost algorithm [10]. In machine learning, a hyperparameter is an important variable that is used as a learner; if it is not set, the value will default. Data has its characteristics, so it is very necessary to set hyperparameters and on the other hand, the model from XGBoost has more than 30 hyperparameters and how to optimize hyperparameters is very important, so it is very important to adjust the hyperparameters [11]. As a result, this study will compare two hyperparameter tuning methods: grid search and random search. It is hoped that by doing a comparison, we can find the best value for the hyperparameter.

Previous studies have used XGBoost with grid search tuning in the case, predicting dementia risk with an accuracy of 85.61% [11]. Then, according to Zhang's research [12], the use of grid search on XGBoost is successful in handling the classification of a problem. Whereas in other studies, random search can produce better models with a fast process [13] and on average, random search on XGBoost can produce sufficient results in some cases [10]. Based on these results the authors want to compare which method is better in predicting CRF between grid search and random search for hyperparameter tuning on XGBoost.

In addition to hyperparameter tuning, the pre-processing process can have an impact on an algorithm's classification results. The two pre-processing stages will then be compared. To handle data imbalances, SMOTE and random oversampling will be compared at the oversampling process stage. To deal with unwanted noise, min-max normalization and z-score normalization are used during the normalization stage. With the hyper parameter tuning and oversampling and normalization processes, it is hoped that XGBoost can classify CRF detection more accurately and optimally by looking at the accuracy and f-measure results.

Several previous studies with the same dataset. Sinha [14] conducted a study by implementing a new decision support system to predict CRF. The algorithms used to classify are SVM and KNN. The accuracy value produced by SVM is 73.75% while

KNN is 78.75%. SVM and KNN produce f-measure values of 0.6670 and 0.8090. Alassaf [15] conducted a study by comparing the ANN, SVM, KNN, Naïve Bayes algorithms. For each algorithm, an optimization strategy is carried out and a feature selection method is added. The results obtained are ANN, SVM, and Naïve Bayes get the best results with an accuracy value of 98% and f-measure 0.98167. Meanwhile, KNN got an accuracy value of 93.90% and f-measure 0.9457.

This research is structured as follows: Section 2 discusses the methods used in this study in detail. Section 3 describes the results and discussion of this research and the conclusions are presented in section 4.

## 2. Methodology

### 2.1 Data collection

The initial stage of the research was data collection. The CRF data obtained had a total of 400 data with 25 attributes. Descriptions of the 25 attributes can be seen in Table 1.

The data used was valid data obtained from the UCI Machine Learning Repository. The data were prepared by L. Jerlin Rubini (researcher) from the University of Alagappa, obtained from Dr. P. Soundarapandian. M.D., D.M. (Senior Nephrologist Consultant at Apollo India Hospital) and under the guidance of Dr. P. Eswaran (Assistant Professor) from the Department of Computer Science and Engineering, Alagappa University, India. The data employed was data on kidney failure patients in India.

### 2.2 Data preprocessing

#### 2.2.1. Data cleaning

In the CRF dataset, there were quite a lot of incomplete data. All existing attributes had incomplete data. Attributes with at least incomplete data were appetite, pedal edema, and anemia, which were only one for each attribute. Meanwhile, attributes with incomplete data were primarily red blood cells, with 152 incomplete data. Total complete data in the dataset was 158 data, and incomplete data was 242 data. Therefore, a quality data cleaning process was needed so that the data could be improved.

Data cleaning is a process used to overcome inaccurate and incomplete data but not to reduce data shortages and improve data quality [16]. There are three ways to overcome data shortages, especially in incomplete data:

Table 1. Variables and data descriptions for the attributes of chronic renal failure

Variable	Attribute	Description
X1	Age	Patient age (years)
X2	Blood Pressure	Patient blood pressure (mm/Hg)
X3	Specific Gravity	The specific gravity of the patient's urine
X4	Albumin	Protein in the patient's blood
X5	Sugar	The patient's blood sugar level
X6	Red Blood Cells	Red blood cells in patients
X7	Pus Cell	The content of white blood cells in the patient's urine
X8	Pus Cell Clumps	Clumps of pus cells in the patient's urine
X9	Bacteria	Groups of organisms that do not have a cell nucleus membrane present in the patient's body
X10	Blood Glucose Random	Random blood sugar check in patients (mgs/dl)
X11	Blood Urea	Urea content in blood (mgs/dl)
X12	Serum Creatinine	The content of creatinine in the blood through examination of the patient's urine (mgs/dl)
X13	Sodium	The content of sodium in the blood through examination of the patient's urine (mgs/dl)
X14	Potassium	The content of potassium in the blood through examination of the patient's urine (mgs/dl)
X15	Hemoglobin	Protein content that carries oxygen in the blood (gms)
X16	Packed Cell Volume	Examination of the percentage of red blood cells to blood volume
X17	White Blood Cell Count	The number of white blood cells in the patient's blood (cells/cmm)
X18	Red Blood Cell Count	The number of white blood cells in the blood (millions/cmm)
X19	Hypertension	High blood pressure in patients
X20	Diabetes Mellitus	A condition in which the pancreas cannot produce the hormone insulin, causing high blood sugar levels in patients
X21	Coronary Artery Disease	A condition in which plaque builds up in the arteries of the heart in patients
X22	Appetite	The patient's appetite
X23	Pedal Edema	A condition in which there is a buildup of fluid in the area of the patient's feet
X24	Anemia	A condition in which there is a lack of red blood cells or non-functioning red blood cells in the patient's body
Y	(target) Classification	Results where the patient was not diagnosed or diagnosed with chronic kidney failure

1. Ignoring incomplete data
2. Filling in incomplete data
3. Deleting incomplete data

In this study, the method utilized in overcoming incomplete data was filling in incomplete data using the mean method for numerical data and the modus method for nominal data.

### 2.2.2. Synthetic minority oversampling technique (SMOTE)

After obtaining complete data, an examination of the data balance was carried out. The difference between those with CRF and those without CRF was more than 50%. Thus, it was necessary to do sampling, namely oversampling. The oversampling

method increases the number of minor class data so that the amount of data can be balanced.

One of the oversampling methods that can be used is the Synthetic Minority Oversampling Technique (SMOTE). The SMOTE method adds the amount of data from the minor class using the neighbor approach to make it equal to the major class [17]. In the case of credit scoring classification, the SMOTE method results in an accuracy value of 81.69%, which is relatively high; this value is better than other methods [18]. The formulation of the SMOTE method can be seen in Eq. (1).

$$X' = X_i + (X_k - X_i) \gamma \quad (1)$$

### 2.2.3. Random oversampling

The following sampling method used was random oversampling. The random oversampling method randomly increases the number of data from the minor class by doubling the data for the minor class, which in previous studies could improve the prediction performance [19]. The way random oversampling works is:

1. Choosing data from the minor class randomly
2. After obtaining data from the minor class, the data will be duplicated.
3. The iteration will continue until the minor class data reaches 50% of the dataset.

### 2.2.4. Data splitting

The next step was splitting the CRF dataset. The splitting process was done by dividing the data into two parts. The first data is training data used to produce a data model for classification. Meanwhile, the second data is test data used to evaluate a data model [20]. In this study, the distribution for the dataset was 70% for training data and 30% for test data purposes.

### 2.2.5. Min-max normalization

The next step was to perform data normalization on the training data. One of the techniques used for normalization is the min-max normalization technique. The min-max normalization method is a normalization technique used to change the data scale from a range to a new range, which in previous studies on the effect of normalization on classification got the best results with 96% accuracy [21]. The min-max normalization technique is shown in Eq. (2).

$$X' = \frac{X - x_{min}}{x_{max} - x_{min}} \quad (2)$$

### 2.2.6. Z-score normalization

The following technique utilized was the z-score normalization technique. This technique normalizes the standard deviation and means of the data. Based on previous research, namely classifying protein crystallization images, the z-score method has succeeded in increasing the accuracy results [22]. The z-score normalization technique can be seen in Eq. (3).

$$X' = \frac{X - \mu_X}{\sigma_X} \quad (3)$$

## 2.3 Data processing

### 2.3.1. Extreme gradient boosting (XGBoost)

XGBoost is an advanced Gradient Boost system designed to be lightweight, efficient, and flexible [20-21]. XGBoost can achieve promising results on a wide variety of classification data. It is reinforced by the many competitions in Kaggle won using XGBoost in the classification process [23].

The basic concept of boosting is to build a more accurate model by combining hundreds of simple trees with low accuracy, where each iteration will produce a new tree for the model. Next, the thing to note is the complexity of the tree. The complexity of the tree will affect the results [12]. The complexity of each tree was formulated in Eq. (4):

$$\Omega(f) = \gamma T + 0.5\lambda \sum_{j=1}^T \omega_j^2 \quad (4)$$

There are many ways to determine the complexity, and they work well [12]. Furthermore, the XGBoost function was formulated in Eq. (5):

$$obj = \sum_{j=1}^T [G_j \omega_j + 0.5(H_j \lambda) \omega_j^2] + \gamma T \quad (5)$$

## 2.4 Hyperparameter tuning

The optimal hyperparameter search can be done using tuning to learn an algorithm based on existing data. In some cases, in supervised learning, the optimal performance of an algorithm depends on hyperparameters [24]. The tuning techniques used in this research were grid search and random search, which would compare which tuning method was better to find the optimal hyperparameter.

### 2.4.1. Grid search

In fact, grid search is an exhaustive search based on subsets, whose hyperparameters are determined by using a lower limit, an upper limit, and the number of steps [25]. The grid method will look for all possibilities by preparing a grid, which will then be evaluated to get the best value among all grids, and all steps are carried out systematically [26]. The grid search method has the advantage of executing data with high accuracy [27]. The way grid search works are as follows:

1. Initializing all values of the parameter
2. Looping the combination of all parameter values
3. Conducting training using XGBoost on training data
4. Evaluating the resulting classifications with test data
5. Storing the best value from the classification result and the best parameter value combination.

### 2.4.2. Random search

The random search method randomly tries several predetermined combinations, and then the hyperparameters are evaluated, then the best results are taken [12, 14]. Random search is efficient and can process data with large dimensions well [10]. The workings of random search can be seen as follows:

1. Initiating the number of iterations of the parameter combination
2. Initializing all values of the parameter
3. Iterating random combinations of parameter values based on the number of iterations
4. Conducting training using XGBoost on training data
5. Evaluating the resulting classifications with test data
6. Storing the best value from the classification result and the best parameter value combination

## 2.5 Evaluation model

### 2.5.1. Confusion matrix

After data processing was done, the next step was to evaluate the model. An evaluation model needs to be done to see the accuracy value and f-measure value of an algorithm in a case. The method employed to determine the accuracy and f-measure in this study was the Confusion Matrix. Confusion Matrix is used to obtain information related to actual and predicted information [28]. An algorithm can be known for its accuracy value by looking at the percentage of the correct prediction value of an algorithm [28] and the f-measure value by weighting the combination of precision, which is the positive predictive value with recall or the correct prediction level [29]. The formulation to determine the accuracy value can be seen in Eq. (6) and f-measure can be seen in Eq. (7).

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \times 100\% \quad (6)$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

### 2.5.2. Cross Validation

Cross-validation is a computer-intensive technique used to evaluate the performance of an algorithm model and the prediction error using all available examples as training and testing examples [17, 28, 29]. K-Fold Cross Validation was the method used in this study. K-fold trains K times of data and leaves  $\frac{1}{K}$  as the test data [30], which then the model

results are the average results based on the training process [17].

## 3. Results and discussion

### 3.1 Data cleaning

In this study, the dataset used was data on kidney failure. The dataset's number was 400 data, with 24 input variables and one target variable. The dataset was analyzed to see the data completeness. After being analyzed, there were 242 incomplete data. Therefore, data cleaning was needed. In the data cleaning process, empty data were filled using the mean method for numerical data and the modus method for categorical data.

### 3.2 Oversampling

Furthermore, an analysis of the amount of target data was carried out. The number of patients with kidney failure was 250 patients with a percentage of 62.25%, and the number of patients who did not suffer from kidney failure was 150 patients with a percentage of 37.75%. Based on this, there was an imbalance of data, so a sampling using the oversampling method was needed. The oversampling methods used in this study were the SMOTE method and random oversampling. After oversampling, there was an increase in the number of data by 100 data so that the amount of data in the dataset was 500 data.

### 3.3 Data splitting and normalization

Next, the data splitting process was conducted by dividing 70% of the data into training data with 350 data and 30% into test data with a total of 150 data. In this dataset, there was too extensive a range of values between one attribute and another. Thus, a normalization method was needed. The normalization methods used were the min-max normalization method and the z-score normalization method.

### 3.4 Hyperparameter tuning on extreme gradient boosting (XGBoost)

In the hyperparameter tuning process, the parameter value range should first be determined.

Table 2. Parameters and parameter value range

Parameter	Parameter Value Range
learning_rate	[0.1, ..., 0.3]
gamma	[0.1, ..., 0.9]
max_depth	[1, 3, 5]
subsample	[0.1, ..., 0.9]
colsample_bytree	[0.1, ..., 0.9]

Table 3. Best parameter combination value

Data	Parameter	Best parameter value	
		Grid search	Random search
Without min-max normalization and SMOTE oversampling	learning_rate	0.03	0.03
	gamma	0.1	0.3
	max_depth	3	3
	subsample	0.9	0.8
	colsample_bytree	0.3	0.1
Min-max normalization	learning_rate	0.03	0.03
	gamma	0.1	0.3
	max_depth	3	3
	subsample	0.9	0.8
	colsample_bytree	0.3	0.1
Z-score normalization	learning_rate	0.03	0.03
	gamma	0.1	0.3
	max_depth	3	3
	subsample	0.9	0.8
	colsample_bytree	0.3	0.1
Min-max normalization and SMOTE oversampling	learning_rate	0.01	0.03
	gamma	0.1	0.3
	max_depth	5	3
	subsample	0.7	0.8
	colsample_bytree	0.1	0.1
Min-max normalization and random oversampling	learning_rate	0.03	0.04
	gamma	0.1	0.3
	max_depth	5	5
	subsample	0.9	0.8
	colsample_bytree	0.2	0.2
Z-score normalization and SMOTE oversampling	learning_rate	0.04	0.04
	gamma	0.1	0.3
	max_depth	3	5
	subsample	0.9	0.8
	colsample_bytree	0.2	0.2
Z-score normalization and random oversampling	learning_rate	0.03	0.04
	gamma	0.1	0.3
	max_depth	5	5
	subsample	0.9	0.8
	colsample_bytree	0.1	0.2

There were five parameters to be tuned. The tuned parameters were: learning\_rate, gamma, max\_depth, subsample, colsample\_bytree. Learning\_rate and gamma have a strong enough influence in influencing the model to get the optimal value. Learning\_rate influences in dealing with overfitting, while gamma

affects reducing loss to determine the leaf node of a tree [9, 20]. Then, max\_depth determines the depth of the tree, while colsample\_bytree is the ratio of the attributes when constructing each tree [12] and subsample handles regularization. The range of values and parameters performed by tuning can be

Table 4. Best accuracy and f-measure values from hyperparameters

Data	Metrics	Best accuracy and f-measure values from hyperparameters	
		Grid search	Random search
Without normalization and oversampling	Accuracy	98.75%	96.78%
	F-Measure	0.9887	0.9772
Min-max normalization	Accuracy	98.75%	96.78%
	F-Measure	0.9887	0.9772
Z-score normalization	Accuracy	98.75%	96.78%
	F-Measure	0.9887	0.9772
Min-max normalization and SMOTE oversampling	Accuracy	98.85%	98.28%
	F-Measure	0.9913	0.9829
Min-max normalization and random oversampling	Accuracy	99.28%	98.57%
	F-Measure	0.9942	0.9858
Z-score normalization and SMOTE oversampling	Accuracy	98.75%	98.57%
	F-Measure	0.9913	0.9842
Z-score normalization and random oversampling	Accuracy	99.28%	98.57%
	F-Measure	0.9942	0.9885

seen in Table 2.

After determining the range of parameter values, hyperparameter tuning was performed by comparing the data that was only done data cleaning without normalization, then normalized data, and normalized and oversampled data. The search for the most optimal parameter values and the hyperparameters' best accuracy and f-measure values utilized the 5-fold cross-validation model. The results of the most optimal parameter values found by grid search and random search can be seen in Table 3.

Based on the Table 3, the combination of hyperparameter values obtained tended to be the same in the random search method. However, using the grid search method, the combination changed when the data were normalized and oversampled. The combination of hyperparameter values obtained by grid search was more diverse.

Based on the parameter combination values found by grid search and random search using 5-fold cross-validation, the accuracy and f-measure results from hyperparameters obtained from the comparison of each data preprocessed using the XGBoost classification algorithm can be seen in Table 4.

Based on the results, regarding the accuracy value with the grid search method and preprocessing data min-max normalization and z-score and oversampling random oversampling, the best accuracy and f-measure results were obtained with scores of 99.28% and 0.9942, respectively. The best combination of hyperparameters produced on min-max normalized data and random oversampling was

learning\_rate = 0.03, gamma = 0.1, max\_depth = 5, subsample = 0.9, and colsample\_bytree = 0.2. Meanwhile, the z-score data normalization and random oversampling resulted in learning\_rate = 0.03, gamma = 0.1, max\_depth = 5, subsample = 0.9, and colsample\_bytree = 0.1 as the best hyperparameter combination. Based on these results, the min-max and z-score normalization processes combined with random oversampling could produce the best accuracy and f-measure values. On the other hand, the combination of preprocessing normalization and oversampling made the random search method increase by 2%. It was because the increase in training data made the model able to learn more and more data. Based on this, it could be concluded that the grid search method could produce a combination of hyperparameters and accuracy and f-measure result better than random search.

### 3.5 Comparative analysis

At the comparative analysis stage, a model evaluation was conducted to predict CRF by comparing the model used in this study with several other methods. The method used in this research is XGBoost which is carried out by a hyperparameter tuning process to find the best hyperparameter values which in Table 4 get the best results using a grid search. The method is combined with the pre-processing process based on the results in Table 4, namely data cleaning, and then oversampling is carried out using random oversampling and then

Table 5. Model evaluation results

Model	Accuracy (%)	F-Measure
SVM [14]	73.75	0.6670
KNN [14]	78.75	0.8090
ANN, Naïve Bayes, and SVM [15]	98.00	0.98167
KNN [15]	93.9	0.9457
XGBoost [9]	99.29	0.99
<b>Proposed Model (XGBoost with hyperparameter tuning grid search and pre-processing random oversampling, and z-score normalization)</b>	<b>99.33</b>	<b>1.0</b>

normalization, where the two methods get the same results. However, to get maximum results, the z-score normalization will be used to normalize the data. [22]. The proposed method in this study will be compared with other methods that have been used previously using a CRF disease dataset, SVM [14], KNN [14], ANN [15], Naïve Bayes [15], and XGBoost [9]. Each method will compare the accuracy and f-measure values. The results of the comparison can be seen in Table 5.

The SVM method classifies by maximizing the margin between two support vectors and KNN and then classifying based on the highest frequency [11]. Naïve Bayes is a method based on Bayes' Theory that calculates the probability with a strong assumption of independence [33]. ANN is a machine learning algorithm that simulates the properties of biological neural networks [15].

According to table 5, the model used in this study produces the best results, with an accuracy value of 99.33% and an f-measure value of 1.0. It produces better accuracy and f-measure results than the previous methods when the hyperparameter tuning grid search process is combined with pre-processing random oversampling and z-score normalization. In the comparison in Table 5, SVM [14] gets the lowest results with an accuracy value of 73.75% and an f-measure value of 0.6670. Based on these results, it can be concluded that the model in this study can help predict CRF well and produce better performance than other methods.

## 4. Conclusion

CRF is one of the diseases of global concern. So we need an algorithm to help detect CRF. One of them is XGBoost with the help of the hyperparameter tuning and pre-processing process. In this study, we compared two tuning methods which include grid search and random search to help predict CRF disease, and the grid search method could find the best hyperparameter combination and produced the best value, with an accuracy value of 99.28% and an f-measure of 0.9942. The normalization and oversampling processes also play a significant role in obtaining these results. In this study, the use of the z-score method combined with random oversampling obtain the best results. The results of the proposed method are compared with previous studies get the best results with an accuracy value of 99.33% and f-measure 1.0. Based on these results, it can be concluded that XGBoost with hyperparameter tuning combined with random oversampling and z-score normalization can help detect CRF very well.

There are still many shortcomings in this research, so further development is needed. First, due to the limitations of the author's knowledge base and time, this research has many shortcomings. Furthermore, the search for combinations of hyperparameter values in the grid search takes a long time. It is hoped that in future research, a combination of methods can be used to reduce computational time so that the grid search method can perform processing with good results and in a faster time.

## Conflict of interest

The authors declare no conflict of interest.

## Author contributions

Conceptualization, Dimas Aryo Anggoro and Salsa Sasmita Mukti; methodology, Dimas Aryo Anggoro and Salsa Sasmita Mukti; software, Salsa Sasmita Mukti; validation, Salsa Sasmita Mukti; formal analysis, Dimas Aryo Anggoro and Salsa Sasmita Mukti; investigation, Salsa Sasmita Mukti; resources, Salsa Sasmita Mukti; data curation, Salsa Sasmita Mukti; writing—original draft preparation, Salsa Sasmita Mukti; writing—review and editing, Dimas Aryo Anggoro; visualization, Salsa Sasmita Mukti; supervision, Dimas Aryo Anggoro; project administration, Dimas Aryo Anggoro; funding acquisition, Dimas Aryo Anggoro.

## References

- [1] R. Devika, S. V. Avilala, and V.



- Subramaniaswamy, “Comparative study of classifier for chronic kidney disease prediction using naive bayes, KNN and random forest”, In: *Proc. of 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 679–684, 2019.
- [2] A. K. Singh, Y. M. Farag, B. V. Mittal, and K. K. Subramanian, “Epidemiology and risk factors of chronic kidney disease in India – results from the SEEK (Screening and Early Evaluation of Kidney Disease) study”, *BMC Nephrol*, Vol. 14, No. 114, pp. 1-10, 2013.
- [3] V. Jha, G. G. Garcia, K. Iseki, Z. Li, S. Naicker, B. Plattner, R. Saran, A. Y. M. Wang, and C. W. Yang, “Chronic kidney disease: Global dimension and perspectives”, *The Lancet*, Vol. 382, No. 9888, pp. 260-272, 2013.
- [4] M. Tandil, A. Mongan, and F. Manoppo, “Hubungan Antara Derajat Penyakit Ginjal Kronik Dengan Nilai Agregasi Trombosit Di Rsup Prof. Dr. R. D. Kandou Manado”, *e-Biomedik*, Vol. 2, No. 2, pp. 509-513, 2014.
- [5] M. M. Rajapurkar, G. T. John, A. L. Kirpalani, G. Abraham, S. K. Agarwal, A. F. Almeida, S. Gangl, A. Gupta, G. Modi, D. Pahari, R. Pisharody, J. Prakash, A. Raman, D. S. Rana, R. K. Sharma, R. N. Sahoo, V. Sakhuja, R. R. Tatapudi, and V. Jha, “What do we know about chronic kidney disease in India: First report of the Indian CKD registry”, *BMC Nephrol*, Vol. 13, No. 1, pp. 1-8, 2012.
- [6] L. Zhang, F. Wang, W. Wang, B. Liu, and J. Liu, “Prevalence of chronic kidney disease in China: A cross-sectional survey”, *The Lancet*, Vol. 379, No. 9818, pp. 815-822, 2012.
- [7] S. Wongthanavasul, “A Bayesian Belief Network Model for Breast Cancer Diagnosis”, In: *Operations Research Proceedings 2010*, pp. 3-8, 2011.
- [8] D. Chicco and G. Jurman, “Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone”, *BMC Medical Informatics and Decision Making*, Vol. 20, No. 1, pp. 1-16, 2020.
- [9] N. V. G. Raju, K. P. Lakshmi, K. G. Praharshitha, and C. Likhitha, “Prediction of chronic kidney disease (CKD) using Data Science”, In: *Proc. of 2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 642–647, 2019.
- [10] S. Putatunda and K. Rama, “A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost”, In: *Proc. of the 2018 International Conference on Signal Processing and Machine Learning*, pp. 6–10, 2018.
- [11] S. E. Ryu, D. H. Shin, and K. Chung, “Prediction Model of Dementia Risk Based on XGBoost Using Derived Variable Extraction and Hyper Parameter Optimization”, *IEEE Access*, Vol. 8, pp. 177708-177720, 2020.
- [12] L. Zhang and C. Zhan, “Machine Learning in Rock Facies Classification: An Application of XGBoost”, In: *Proc. of International Geophysical Conference, Qingdao, China*, pp. 1371–1374, 2017.
- [13] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization”, *Journal of Machine Learning Research*, Vol. 13, No. 2, pp. 281–305, 2012.
- [14] P. Sinha and P. Sinha, “Comparative Study of Chronic Kidney Disease Prediction using KNN and SVM”, *International Journal of Engineering Research and Technology*, Vol. 4, No. 12, pp. 608–612, 2015.
- [15] R. A. Alassaf, K. A. Alsulaim, N. Y. Alroomi, N. S. Alsharif, M. F. Aljubeir, S. O. Olatunji, A. Y. Alahmadi, M. Imran, R. A. Alzahrani, and N. S. Alturayef, “Preemptive Diagnosis of Chronic Kidney Disease Using Machine Learning Techniques”, In: *Proc. of 2018 International Conference on Innovations in Information Technology (IIT)*, pp. 99–104, 2018.
- [16] K. Natarajan, J. Li, and A. Koronios, “Data mining techniques for data cleaning”, In: *Proc. of the 4th World Congress on Engineering Asset Management*, pp. 796–804, 2009.
- [17] M. Syukron, R. Santoso, and T. Widiharid, “Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data”, *Jurnal Gaussian*, Vol. 9, No. 3, pp. 227-236, 2020.
- [18] O. Heranova, “Synthetic Minority Oversampling Technique pada Averaged One Dependence Estimators untuk Klasifikasi Credit Scoring”, *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, Vol. 3, No. 3, pp. 443-450, 2019.
- [19] Y. Kamei, A. Monden, S. Matsumoto, T. Kakimoto, and K. Matsumoto, “The Effects of Over and Under Sampling on Fault-prone Module Detection”, In: *Proc. of First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, pp. 196–204, 2007.
- [20] D. A. Anggoro and P. I. Rahmatullah, “The Implementation of Subspace Outlier Detection in K-Nearest Neighbors to Improve Accuracy in Bank Marketing Data”, *International Journal of*

- Emerging Trends in Engineering Research*, Vol. 8, No. 2, pp. 545-550, 2020.
- [21] N. Chamidah, Wiharto, and U. Salamah, "Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropagasi Gradient Descent Adaptive Gain (BPGDAG) untuk Klasifikasi", *Jurnal ITSMART*, Vol. 1, No. 1, pp. 28-33, 2012.
- [22] I. Dinc, M. Sigdel, S. Dinc, M. S. Sigdel, M. L. Pusey, and R. S. Aygun, "Evaluation of normalization and PCA on the performance of classifiers for protein crystallization images", In: *Proc. of IEEE SOUTHEASTCON 2014*, pp. 1-6, 2014.
- [23] Y. Xia, C. Liu, Y. Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyperparameter optimization for credit scoring", *Expert Systems with Applications*, Vol. 78, pp. 225-241, 2017.
- [24] P. Probst, M. N. Wright, and A. L. Boulesteix, "Hyperparameters and tuning strategies for random forest", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 9, No. 3, pp. 1-15, 2019.
- [25] I. Syarif, A. P. Bennett, and G. Wills, "SVM parameter optimization using grid search and genetic algorithm to improve classification performance", *Telkomnika*, Vol. 14, No. 4, pp. 1502-1509, 2016.
- [26] M. Ataei and M. Osanloo, "Using a combination of genetic algorithm and the grid search method to determine optimum cutoff grades of multiple metal deposits", *International Journal of Surface Mining, Reclamation and Environment*, Vol. 18, No. 1, pp. 60-78, 2004.
- [27] T. Xiao, D. Ren, S. Lei, J. Zhang, and X. Liu, "Based on grid-search and PSO parameter optimization for Support Vector Machine", In: *Proc. of the 11th World Congress on Intelligent Control and Automation*, pp. 1529-1533, 2014.
- [28] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem", *Information Sciences*, Vol. 340, pp. 250-261, 2016.
- [29] D. Hand and P. Christen, "A note on using the F-measure for evaluating record linkage algorithms", *Statistics and Computing*, Vol. 28, No. 3, pp. 539-547, 2018.
- [30] Y. Bengio, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation", *Journal of Machine Learning Research* 5, Vol. 5, pp. 1089-1105, 2004.
- [31] T. Fushiki, "Estimation of prediction error by using K -fold cross-validation", *Statistics and Computing*, Vol. 21, No. 2, pp. 137-146, 2011.
- [32] A. Ogunleye and Q. G. Wang, "XGBoost Model for Chronic Kidney Disease Diagnosis", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 17, No. 6, pp. 2131-2140, 2020.
- [33] S. Vijayarani and S. Dhayanand, "Data Mining Classification Algorithms for Kidney Disease Prediction", *International Journal on Cybernetics & Informatics*, Vol. 4, No. 4, pp. 13-25, 2015.