Chronic Kidney Disease Classification Using Bagging and Particle Swarm Optimization Techniques

Suhendro Y. Irianto[1]*, Dephi Linda[2], Immaniar I.M.Rizki[3], Sri Karnila[4], Dona Yuliawati[5]
Dept. of Informatics, Institute Informatics and Business Darmajaya, Bandarlampung, Indonesia[1, 3]
Dept. of Information System, Institute Informatics and Business Darmajaya, Bamdarlampung, Indonesia[2, 4, 5]

***Abstract*—Chronic kidney disease (CKD) is a serious chronic illness without a definitive cure. According to WHO in 2015, 10% of the population suffers from CKD, with 1.5 million patients undergoing global haemodialysis. The incidence of CKD is increasing by 8% annually, ranking it as the 20th highest cause of global mortality. The Random Forest (RF) technique utilizes decision trees as an ensemble model, where class predictions are derived from the combination of results from each tree. The final decision is based on the highest outcome of class predictions generated by each decision tree, employed in this study. In testing, Random Forest with PSO-based Bagging achieved the highest performance with precision of 98.12%, recall of 100.00%, and AUC of 0.999. The Random Forest with PSO-based Bagging model demonstrates high performance in CKD detection, but metrics like precision, recall, and AUC alone do not guarantee clinical applicability. Balancing false positives and negatives is crucial, and its real-world integration should be evaluated to assess its impact on patient outcomes and clinical workflows. Research on predicting chronic kidney disease using the Random Forest algorithm with Bagging based on Particle Swarm Optimization (PSO) indicates that Bagging with PSO feature selection can enhance accuracy and kappa values. These findings contribute to understanding the roles of Bagging and PSO methods in improving the performance of several algorithms, including Random Forest.**

Keywords—Kidney disease; PSO; bagging; Random Forest

I. INTRODUCTION

The World Health Organization (WHO) stated in 2015 that the incidence of CKD reached 10% of the population, and there were 1.5 million CKD patients undergoing haemodialysis (HD) worldwide. This number is expected to increase by 8 percent per year [1]. CKD is a chronic disease with the 20th highest global mortality rate [2].Compared to patients with other conditions, chronic kidney disease (CKD) patients have a mortality rate of 75% and a fivefold risk of hospitalization [3]. This aligns with the increased mortality rate from chronic kidney disease over the past ten years, making it the second highest cause of death worldwide after diabetes [4]. More than 2 million people have been diagnosed with chronic kidney disease (CKD), and only 10% of those two million people receive adequate treatment. Even in the United States, 87.3% of people undergo peritoneal dialysis, and 2.5% receive kidney transplants [5]. Therefore, to treat chronic kidney disease promptly, a method to diagnose the condition is needed [6].

The best accuracy can be obtained by conducting research on the categorization of chronic kidney failure using Particle Swarm Optimization (PSO) and Random Forest optimization. PSO is an optimization technique that, according to previous research, can be used to diagnose disease problems in very large datasets, with PSO optimization achieving the highest accuracy rate of 99.167% [7]. Through research on improving the accuracy of the C4.5 algorithm classification using the bagging technique in heart disease diagnosis, an accuracy rate of 81.84% was obtained [8].

Meanwhile, a study by [9] and proposed FPA-DNN model was evaluated through simulation analysis using the benchmark CKD dataset. The results were analysed from various perspectives and demonstrated the exceptional performance of the FPA-DNN technique, achieving a sensitivity of 98.80%, specificity of 98.66%, accuracy of 98.75%, an F-score of 99%, and a kappa value of 97.33%. Whilst making Random Forest the best algorithm for predicting coronary heart disease [10], [11]. As a result, more research is needed to identify more accurate techniques that offer better diagnostic accuracy. In this case, PSO, Bagging, and Random Forest will be used in the research because other hybrid techniques are needed to optimize the algorithm for diagnosing chronic kidney disease.

According to study [12], Adaptive Backpropagation Neural Network (ABPNN-ANFIS) is then classified using fuzzy logic, which integrates the ABPNN results for enhanced decision-making. It can assist experts in determining the stage `of chronic kidney disease. The Adaptive Neuron Clearing Inference System (ABPNN-ANFIS) was implemented in MATLAB to develop adaptive inverse neural networks. The results indicate that the proposed ABPNN-ANFIS model achieves an efficiency of 98% in terms of accuracy. Another works introduced by study [13] that Deep learning algorithms (DLAs) surpassed the Kidney Failure Risk Equation (KFRE) in predicting the initiation of renal replacement therapy (RRT). The model integrating CNN, LSTM, and ANN layers achieved a ROC-AUC of 0.90, while the standalone CNN reached 0.91. In comparison, both the 4-variable and 8-variable KFRE models attained a ROC-AUC of 0.84. Furthermore, DLAs accurately predicted uncoded renal transplants and identified patients who would require dialysis after five years, demonstrating their ability to capture complex, non-linear patterns.

The problem of classification of chronic kidney disease involves developing a robust and accurate model to identify chronic kidney disease (CKD) in patients based on medical data. CKD is a serious condition that requires early detection to prevent progression to more severe stages. The challenge lies in accurately classifying patients into CKD and non-CKD categories using a large dataset that may contain noisy or imbalanced data.

Therefore, the aim of this research is to develop a robust and accurate model for the classification of chronic kidney disease (CKD) by integrating Bagging and Particle Swarm Optimization (PSO) methods. The objective is to improve the detection and classification of CKD from medical data, ensuring early and reliable diagnosis. By addressing challenges such as noisy or imbalanced data, the research seeks to enhance classification accuracy, minimize false positives and negatives, and contribute to more effective early intervention and treatment of CKD. To address this, the approach combines Bagging, an ensemble method that improves the stability and accuracy of machine learning algorithms by creating multiple versions of a model and averaging their predictions, with Particle Swarm Optimization (PSO), a technique inspired by the social behaviour of birds to optimize the model's parameters. The goal is to enhance classification accuracy, reduce false positives and negatives, and ultimately improve the model's ability to detect CKD, thereby aiding in timely diagnosis and treatment.

II. Methods

The data was processed using RapidMiner, including preprocessing, to prepare it for further data mining operations. Data pre-processing was carried out by handling missing values, as chronic kidney disease (CKD) datasets often contain missing data due to incomplete medical records. Common techniques for handling missing data include mean/mode imputation, K-nearest neighbours (KNN) imputation, or removing records with excessive missing values to preserve data integrity. The dataset is shown in Fig. 2.

A. *Random Forest*

Several decision trees are created using the Random Forest (RF) technique, where each tree is combined and functions as an ensemble model. Each decision tree has class predictions, and choices are arranged based on the highest results [14]. There are several processes involved in using the Random Forest approach, specifically [15]. The process begins with the random sampling stage, where data is drawn with replacement from the training set using a technique known as bootstrapping. Next, during the random subsetting stage, trees are constructed using different variables selected through the optimal random discount process (m < d) based on the available data. These two steps are repeated k times until k trees are randomly generated. Finally, a combined estimate is obtained from the k trees, which can be applied to regression by averaging the results or to classification by taking the majority selected.

The goal of this technique is to build decision trees consisting of root nodes, internal nodes, and leaf nodes using data and attributes randomly. The root node is the top node of the decision tree, and internal nodes are branching nodes that have one input and at least two outputs. Leaf nodes, or terminal nodes, are the final nodes, which only have one input and no outputs. Entropy value calculation uses the formula in Eq. (1).

$$Entropy(y) = -\sum ip\left(\frac{c}{y}\right)\log p(\frac{c}{y}) \qquad (1)$$

The Eq. (1) represents the concept of entropy in information theory, which quantifies the uncertainty or disorder within a probability distribution *yyy*. In this context, entropy is a measure of how unpredictable the outcomes are within the distribution. The equation sums the product of each outcome's probability *p(c/y), p(c/y), p(c/y)* and the logarithm of that probability across all possible outcomes. The negative sign ensures that entropy is a positive value, reflecting the average level of "information" or "uncertainty" inherent in the distribution. When all outcomes are equally likely, the entropy is higher, indicating greater uncertainty. Conversely, when one outcome is much more likely than others, the entropy is lower, signifying less uncertainty. This measure is crucial in various fields, including machine learning, where it helps in decision-making processes, such as determining the most informative feature in decision trees [16].

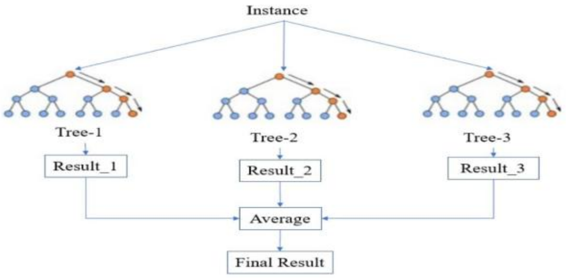


Fig. 1. Simple structure of a Random Forest.

Fig. 1 illustrates a random forest is an ensemble learning technique that enhances predictive accuracy and reduces the risk of overfitting by combining multiple decision trees. In its basic framework, numerous decision trees are constructed using different subsets of the training data and features. Each tree independently generates a prediction, and the final output is determined by aggregating their results—usually through majority voting for classification tasks or averaging for regression tasks. This approach increases robustness and accuracy compared to using a single decision tree, as it reduces variance and mitigates the risk of overfitting. Each sub-tree model performs random sampling with replacement from the training data and ultimately produces an average result from all sub-models [17]. Each sub-model runs in parallel without dependencies. Besides building each tree using different data subsets, random forest differs in how these trees are constructed [18] . In a standard decision tree, each node splits based on the most optimal decision across all variables, minimizing entropy by dividing the dataset represented by the parent node. In contrast, a random forest selects the split point for each node randomly from the best split points within a subset of predictors, [19]. Moreover the study in [9] proposed FPA-DNN model was evaluated using the benchmark CKD dataset. Results confirmed its superior performance, achieving 98.80% sensitivity, 98.66% specificity, 98.75% accuracy, a 99% F-score, and a 97.33% kappa score.

B. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is used by study [20] to model the swarming behaviour of insects, including birds, termites, ants, and bees. The PSO algorithm mimics the social interactions of these animals. Social behaviour includes every action performed by an individual as well as the influence of other group members. For example, the term "particle" describes a flock of birds. With their intelligence, each particle or individual acts in a distributed manner, and their intelligence also affects the behaviour of the aggregate group. Consequently, no matter how far they are from the group, other members can quickly follow if one particle or bird finds the right or shortest path to a food source. The swarm is of a definite or fixed size in multivariate optimization, with each particle starting from a random location in multidimensional space. It is believed that each particle has two characteristics: location and velocity. Each particle in each space remembers its optimal location that emerged or was found concerning the objective function or food source. After providing the information or desired location to other particles, each particle adjusts its position and velocity according to the chosen information position of other particles. For example, the behaviour of birds in a flock. Consequently, the behaviour of a flock of birds will depend on the combination of the following three basic factors: Cohesion, or the ability to fly together; Separation, or not being too close; Alignment, or knowing to head in the same general direction. According to study [21], [22] PSO is designed around the idea that birds, while not explicitly following one another, tend to adjust their paths based on the movements of others when searching for food. Each particle's behaviour is influenced by both its own experience and the collective behaviour of the swarm. This process is repeatedly simulated within a multi-dimensional space, with each iteration gradually steering the particles toward the optimal solution—whether it involves minimizing or maximizing the target function. The iterative process continues until specific convergence criteria are met or the maximum number of iterations is reached [17].

Furthermore the study in [23] explained that particle Swarm Optimization (PSO) is a swarm intelligence-based algorithm used to optimize hyperparameters in machine learning models. Particles, representing candidate solutions, navigate the search space by updating their positions based on their personal best (pBest) and the global best (gBest) solution found by the swarm. This iterative process refines hyperparameter selection, minimizing model error. PSO enhances Bagging by optimizing base learners, sampling ratios, and model parameters, improving ensemble diversity. In Random Forest, it fine-tunes tree-related parameters, balancing bias and variance. By automating hyperparameter tuning, PSO improves model generalization, reduces overfitting, and enhances predictive accuracy efficiently. According to study [24] who described that the velocity update in the Particle Swarm Optimization algorithm, balancing inertia, personal experience, and the global best influence on movement. It is written as Eq. (2).

$$\mathrm{v}_{\mathrm{i}}^{\mathrm{r+1}}=\omega.\mathrm{v}_{\mathrm{i}}^{\mathrm{t}}+c_1.r_1.(pBest_i\text{- }\mathrm{x}_{\mathrm{i}}^{\mathrm{t}})+c_2.r_2.(\mathrm{gBest\text{-}x}_{\mathrm{i}}^{\mathrm{t}} \qquad (2)$$

where:

- v_i^{r+1}: *Velocity of the i*th *particle at iteration r+1.*
- *ω: Inertia weight, controlling the influence of the previous velocity.*
- v_i^t: *Velocity of the iii-th particle at iteration ttt.*
- c_1: *Cognitive acceleration coefficient, influencing personal experience.*
- r_1: *Random factor (uniformly distributed) associated with the cognitive component.*
- $pBest_i$: *Personal best position of the iii-th particle.*
- x_i^t: *Current position of the iii-th particle at iteration ttt.*
- c_2: *Social acceleration coefficient, influencing global experience.*
- r_2: *Random factor (uniformly distributed) associated with the social component.*
- *gBest: Global best position among all particles.*

Study by [25] described that PSO enhances Bagging and Random Forest by optimizing hyperparameters, improving performance and generalization. In Bagging, PSO fine-tunes the number of base learners, data subsampling ratio, and model-specific parameters, boosting ensemble diversity and stability. In Random Forest, it optimizes the number of trees, maximum depth, feature selection, and split criteria, balancing bias and variance. By automating hyperparameter selection, PSO reduces manual effort, making both techniques more efficient and effective for complex predictive tasks.

C. Cross Validation

Cross-validation is one metric for measuring the results of classification algorithms. Meanwhile, K-fold validation is one method to determine the average success rate of a classification system. K-fold validation will randomly shuffle a dataset, allowing the system to be tested on various previously randomized datasets [26], [27]. Furthermore, as stated by study [28] , [29] the purpose of cross-validation is to prevent data from dominating the learning of the classification model. The division of data into the desired n-fold will be used for k-fold validation. For example, if the data s into 5, it will produce 5 data partitions of the same size, such as D1, D2, and D3. After that, the testing and training processes are carried out as many times as the number of folds. The *n* partition data will become the test dataset divided and the training dataset in each ith iteration. The Confusion Matrix contains four combinations of actual and predicted values.

D. Calculating Accuracy

Accuracy is a measure used to evaluate classification models. Simply put, it represents the percentage of predictions made by the model that are correct. As shown in Equation (3-4), accuracy can also be calculated in terms of positives and negatives, [23], [24]. The accuracy in Eq. (3) measures a model's performance by calculating the proportion of correctly predicted positive (TP) and negative (TN) instances out of all predictions.

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

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

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

The classification accuracy value is shown by the TP (True Positive) and TN (True Negative) scores. Generally, classification accuracy is higher with larger TP and TN values. False Positive (FP) occurs when the output prediction label is positive, but the actual value is incorrect. False Negative (FN) occurs when the output prediction label is negative, but the actual result is correct. Moreover, stated that the ratio of related items selected to all selected items in the Confusion Matrix is known as accuracy. Furthermore, accuracy is the degree of conformity between the data expected by the user and the system's response [16]. Eq. (2) measures a model's accuracy in identifying positive instances. It represents the proportion of true positives (TP) out of all instances predicted as positive, including false positives (FP). High precision indicates outcome; the model predicts a positive outcome; it is likely correct. This metric is particularly important in scenarios where false positives are costly or undesirable. The probability of the relevant item being selected is called recall.

Recall is a metric that evaluates a model's effectiveness in identifying all relevant instances within a dataset. It is calculated as the ratio of true positives (TP) correctly identified by the model to the total number of actual positive cases, which comprises both true positives and false negatives (FN). A high recall signifies that the model successfully detects most positive instances making it particularly critical in situations where failing to identify positive cases can have severe consequences, such as in medical diagnostics or fraud detection. While high recall is desirable, it may come at the expense of precision, as the model might also flag more false positives. Balancing recall with precision is essential for achieving overall effectiveness and ensuring that the model performs well across various aspects of its predictions.

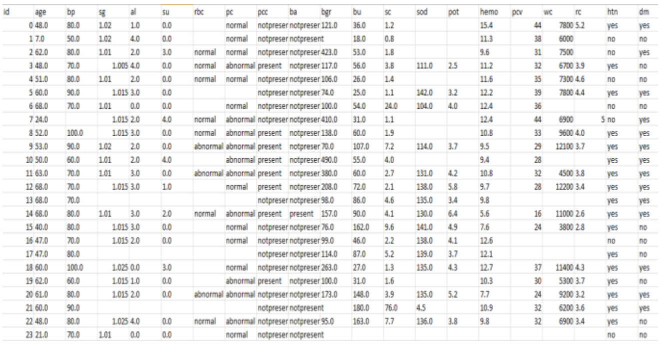
E. Bagging

Introduced by study [30], bagging, also known as bootstrap aggregating, is a classical method for ensemble creation. Although data regression problems may also benefit from its use, classification problems are its primary goal. This is shown by taking multiple samples from the same dataset with replacement through the bootstrap technique. This is useful for generating aggregate predictions because it allows the creation of multiple different trees for the same estimation [31]. The basic principle of the bagging method is to create a new dataset by randomly resampling the original dataset and returning it. Using a random sample of size N with replacement from the training data (bootstrap sample SkS_kSk from DkD_kDk), the [3∣D∣|D|∣D∣. Classification trees with various versions are then created with the new dataset. The final estimate is then produced by combining the classification trees from each version [32] . The final estimate of this method can be produced by voting or averaging for challenges related to regression and classification. This allows multiple samples to be set to be the same [21]. The goal is to generate data subsets using surrogate variables from randomly selected training sets. Essentially, the learning process is trained using each subset of the dataset. As a result, we have a set of different models. By using the average of all predictions from different base learners, the results are more reliable than just using one base learner [33]. The benefit of batch creation is to reduce errors in basic predictors, which may be unstable before specific disturbances, and to provide an estimate of their predictive performance, hampered by the test set or cross-validation estimate [34], [35], [36]. The bagging method consists of two stages. Bootstrapping is the first step, and aggregation is the second. Samples from the available training data are used for the bootstrap stage, and aggregation is the second step.

The dataset contains the following attributes: ID, Age, Blood Pressure (BP), Specific Gravity (SG), Albumin (AL), Sugar (SU), Red Blood Cells (RBC), Pus Cells (PC), Pus Cell Clumps (PCC), Bacteria (BA), Blood Glucose Random (BGR), Blood Urea (BU), Serum Creatinine (SC), Sodium (SOD), Potassium (POT), Hemoglobin (HGB), Packed Cell Volume, White Blood Cell Count (WBC), Red Blood Cell Count (RBC), Hypertension (HTN), Diabetes Mellitus (DM), Appetite (APPET), Pedal Edema (PE), and Anemia (ANE).

III. Results and Discussion

Transforming raw or original data is the initial step in the data mining process. This dataset contains 400 records and 26 attributes, sourced from Kaggle (https://www.kaggle.com/datasets/mahmoudlimam/preprocessed-chronic-kidney-disease-dataset).



id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm
0	48.0	80.0	1.02	1.0	0.0		normal	notpreser	notpreser	121.0	36.0	1.2			15.4	44	7800	5.2	yes	yes
1	7.0	50.0	1.02	4.0	0.0		normal	notpreser	notpresent		18.0	0.8			11.3	38	6000		no	no
2	62.0	80.0	1.01	2.0	3.0	normal	normal	notpreser	notpreser	423.0	53.0	1.8			9.6	31	7500		no	yes
3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpreser	117.0	56.0	3.8	111.0	2.5	11.2	32	6700	3.9	yes	no
4	51.0	80.0	1.01	2.0	0.0	normal	normal	notpreser	notpreser	106.0	26.0	1.4			11.6	35	7300	4.6	no	no
5	60.0	90.0	1.015	3.0	0.0			notpreser	notpreser	74.0	25.0	1.1	142.0	3.2	12.2	39	7800	4.4	yes	yes
6	68.0	70.0	1.01	0.0	0.0		normal	notpreser	notpreser	100.0	54.0	24.0	104.0	4.0	12.4	36			no	no
7	24.0		1.015	2.0	4.0	normal	abnormal	notpreser	notpreser	410.0	31.0	1.1			12.4	44	6900	5	no	yes
8	52.0	100.0	1.015	3.0	0.0	normal	abnormal	present	notpreser	138.0	60.0	1.9			10.8	33	9600	4.0	yes	yes
9	53.0	90.0	1.02	2.0	0.0	abnormal	abnormal	present	notpreser	70.0	107.0	7.2	114.0	3.7	9.5	29	12100	3.7	yes	yes
10	50.0	60.0	1.01	2.0	4.0		abnormal	present	notpreser	490.0	55.0	4.0			9.4	28			yes	yes
11	63.0	70.0	1.01	3.0	0.0	abnormal	abnormal	present	notpreser	380.0	60.0	2.7	131.0	4.2	10.8	32	4500	3.8	yes	yes
12	68.0	70.0	1.015	3.0	1.0		normal	present	notpreser	208.0	72.0	2.1	138.0	5.8	9.7	28	12200	3.4	yes	yes
13	68.0	70.0						notpreser	notpreser	98.0	86.0	4.6	135.0	3.4	9.8				yes	yes
14	68.0	80.0	1.01	3.0	2.0	normal	abnormal	present	present	157.0	90.0	4.1	130.0	6.4	5.6	16	11000	2.6	yes	yes
15	40.0	80.0	1.015	3.0	0.0		normal	notpreser	notpreser	76.0	162.0	9.6	141.0	4.9	7.6	24	3800	2.8	yes	no
16	47.0	70.0	1.015	2.0	0.0		normal	notpreser	notpreser	99.0	46.0	2.2	138.0	4.1	12.6				no	no
17	47.0	80.0						notpreser	notpreser	114.0	87.0	5.2	139.0	3.7	12.1				yes	no
18	60.0	100.0	1.025	0.0	3.0		normal	notpreser	notpreser	263.0	27.0	1.3	135.0	4.3	12.7	37	11400	4.3	yes	yes
19	62.0	60.0	1.015	1.0	0.0		abnormal	present	notpreser	100.0	31.0	1.6			10.3	30	5300	3.7	yes	no
20	61.0	80.0	1.015	2.0	0.0	abnormal	abnormal	notpreser	notpreser	173.0	148.0	3.9	135.0	5.2	7.7	24	9200	3.2	yes	yes
21	60.0	90.0						notpreser	notpresent		180.0	76.0	4.5		10.9	32	6200	3.6	yes	yes
22	48.0	80.0	1.025	4.0	0.0	normal	abnormal	notpreser	notpreser	95.0	163.0	7.7	136.0	3.8	9.8	32	6900	3.4	yes	no
23	21.0	70.0	1.01	0.0	0.0		normal	notpreser	notpresent										no	no

Fig. 2. Chronic kidney disease dataset.

Table I presents the evaluation data that includes performance metrics from four studies using the Random Forest classification algorithm.

TABLE I PERFORMANCE METRICS OF CLASSIFICATION OF THREE ALGORITHMS

Algorithm	Accuracy (%)	Precision (%)	Recall (%)
Random Forest	98.75	98.04	98.67
BNC [37]	96.43	93.02	93.18
KNN+PSO [36]	97.25	N/A	N/A
Fuzzy [38]]	98.28	N/A	N/A

From the evaluation results of the four classification algorithms, Random Forest stands out with the highest accuracy of 98.75%, precision of 98.04%, recall of 98.67%, and AUC of 99.9%. The BNC model, while having a slightly lower accuracy (96.43%), still shows good performance with precision and recall, both reaching 93.02% and 93.18%, respectively. KNN+PSO achieves an accuracy of 97.25% and AUC of 99.9%, but precision and recall information are not available.

Meanwhile, the Fuzzy model achieves a high accuracy of 98.28%, but details on precision, recall, and AUC are not provided. Overall, Random Forest and BNC stand out as good choices with consistent performance, whereas KNN+PSO and Fuzzy require more information for comprehensive evaluation.

Random Forest provides a high combination of accuracy, precision, recall, and AUC, making it a solid choice for classification problems. Although BNC has slightly lower accuracy, it still offers a good balance between precision and recall. KNN+PSO shows good results in terms of accuracy and AUC, but the lack of information on precision and recall limits accuracy, requires additional information to measure its prediction quality. Therefore, the selection of an algorithm should be based on the specific needs of the application, and further evaluation, especially on precision and recall, can provide deeper insights into the model's ability to handle positive and negative cases.

Random Forest demonstrates superior performance with high levels of accuracy, precision, and recall, and an AUC of 0.999, showcasing its skill in classifying data. The performance evaluation of the Random Forest, Naïve Bayes, and k-NN algorithms using the bagging method shown in Table II describes the performance metrics of the three different classification algorithms. Table III shows the performance of the Random Forest algorithm after applying the bagging method. Random Forest with Bagging and Random Forest alone yield the same results, with accuracy, precision, and recall each at 98.75%, and AUC at 0.999. k-NN with Bagging shows improved performance compared to k-NN alone, with an accuracy of 74.25%, precision of 62.06%, recall of 83.33%, and AUC of 0.821. Meanwhile, Naïve Bayes with Bagging shows a decrease in performance, with an accuracy of 94.25%, precision of 87.21%, recall of 100.00%, and AUC of 0.996.

The Random Forest model achieves high accuracy (98.75%), precision (98.04%), and recall (98.67%) in predicting outcomes, but its interpretability in medical applications remains a challenge. Unlike simpler models, Random Forest functions as an ensemble of decision trees, making it difficult to explain individual predictions. In healthcare, transparency is crucial for clinical trust and decision-making. Black-box models like Random Forest can hinder adoption due to limited explainability. However, techniques such as feature importance analysis, SHAP, and LIME can help interpret predictions by identifying key influencing factors, enabling clinicians to better understand, validate, and apply the model's outputs effectively.

The table presents the accuracy, precision, and recall of various classification algorithms for chronic kidney disease (CKD) diagnosis, emphasizing their strengths and potential misclassification errors. Among them, Random Forest (RF) achieves the highest accuracy at 98.75%, with a low false positive rate (precision: 98.04%) and low false negative rate (recall: 98.67%), making it the most reliable model. The Bayesian Network Classifier (BNC) has a lower accuracy (96.43%) and higher misclassification rates, as indicated by its 93.02% precision and 93.18% recall, making it less reliable for high-risk CKD detection. K-Nearest Neighbours with Particle Swarm Optimization (KNN+PSO) achieves an accuracy of 97.25%, but the lack of precision and recall data makes error assessment challenging. Similarly, the Fuzzy Logic model has a slightly lower accuracy than RF (98.28%), but without precision and recall metrics, misclassification errors remain unclear. Overall, Random Forest emerges as the most effective model due to its high accuracy and well-balanced false positive and false negative rates.

The Random Forest algorithm demonstrates outstanding performance across key evaluation metrics, achieving an accuracy of 98.75%, which indicates that 98.75% of instances are classified correctly and reinforces the model's reliability. Its precision of 98.04% means that when the model predicts a positive outcome, it is accurate 98.04% of the time, leading to a low false positive rate, while a recall of 98.67% shows it accurately identifies 98.67% of actual positive cases, reflecting a low false negative rate. These metrics highlight the exceptional balance between precision and recall in the Random Forest algorithm, making it a reliable choice for classification tasks. Nevertheless, for real-world applications, it is crucial to evaluate the dataset's size and diversity, as validating the model on larger and more varied datasets would confirm its robustness and scalability. Incorporating additional metrics like the F1-score and AUC-ROC could also provide deeper insights into its overall effectiveness. However, the model's complexity, as it operates as an ensemble of trees, may hinder interpretability, particularly in medical settings where clear decision-making is essential. Furthermore, the lack of statistical significance tests in the results makes it difficult to determine if the performance differences among algorithms are meaningful, leaving reported improvements unvalidated.

In addition, computational aspects and interpretability also need to be considered when choosing an algorithm. While Random Forest and BNC show good performance, they have high model complexity, which can be a consideration in terms of model readability. On the other hand, KNN+PSO and Fuzzy, although providing good results in some metrics, lack information on precision and recall, as well as AUC, which can be a hindrance to a deep understanding of their performance. It is important to continue exploring and understanding the characteristics of each algorithm and make necessary adjustments according to the specific needs of the application. A holistic evaluation, including an analysis of computational properties and interpretability, will help in selecting the most suitable algorithm for the given classification task. In conclusion, the selection of a classification algorithm should consider various factors, including accuracy, precision, recall, AUC, as well as computational and interpretability aspects, to ensure it fits the specific needs of a large-scale application.

The performance of the four classification algorithms shows that Random Forest delivers excellent results with an accuracy of 98.75%, precision of 98.04%, recall of 98.67%, and AUC of 99.9%. The BNC algorithm, although with slightly lower accuracy at 96.43%, still shows solid performance with precision and recall each reaching 93.02% and 93.18%, and an AUC of 93.2%. KNN+PSO achieves an accuracy of 97.25% and an AUC of 99.9%, but precision and recall information is not available. Meanwhile, the Fuzzy algorithm reaches a high accuracy of 98.28%, but information on precision, recall, and AUC cannot be evaluated based on the provided data. Generally, Random Forest and BNC show consistent and reliable performance, while KNN+PSO and Fuzzy require more information for a thorough evaluation. It should be noted that the appropriate algorithm choice should be based on the specific application needs and desired analysis goals.

Each algorithm has its strengths and weaknesses. Random Forest stands out in accuracy and ability to handle model complexity, while BNC shows a good balance between precision and recall. KNN+PSO provides high accuracy and good AUC, but the unavailability of information on precision and recall can be a limitation in understanding the overall model performance. On the other hand, the Fuzzy algorithm provides high accuracy, but the lack of other information makes performance interpretation more difficult. Algorithm selection should be carefully considered based on the dataset characteristics, sample size, and analysis objectives. Moreover, it is important to consider the trade-offs between accuracy, precision, and recall depending on the application needs. A holistic evaluation and deep understanding of performance metrics will help researchers and practitioners make the right decisions in choosing the classification algorithm that suits the context. Continuing to explore and understand the latest developments in this field is also important to ensure that the applied solutions remain relevant and effective over time.

The research findings on chronic kidney disease prediction using the Random Forest algorithm with a Bagging approach based on Particle Swarm Optimization (PSO) have been presented. Therefore, it can be concluded that the use of the Bagging method with Particle Swarm Optimization (PSO) for feature selection can improve accuracy and kappa values across several algorithms, including Random Forest, Naïve Bayes, and k-NN. In testing, Random Forest with PSO-based Bagging achieved the highest performance with a precision of 98.12%, recall of 100.00%, and an AUC of 0.999. This indicates that the model built has a high level of agreement between the predictions made by the model and the actual values in the test data. In other words, the higher the AUC value, the better the model is at predicting the target class or variable. The research still requires further development to improve its performance. Future research and development can be conducted using more appropriate attributes and incorporating digital image objects. When considering the performance of classification algorithms, it is important to note that a deep understanding of the dataset's characteristics and the application context is key. Random Forest demonstrated impressive capabilities in handling complexity and providing accurate predictions. BNC, with a balance between precision and recall, is suitable for situations where it is important to detect most positive instances without compromising overall accuracy. KNN+PSO, although yielding good results, requires further information to fully understand its ability to handle both positive and negative cases. The Fuzzy algorithm, while having high accuracy, requires better interpretability through additional information.

A dataset of 400 records may appear limited, however it can still be sufficient depending on the problem's complexity, the data's quality, and the consistency of patterns within the dataset. A well-curated and representative dataset can offer meaningful insights into the model's performance. Furthermore, if the model exhibits stable and consistent results through cross-validation or other robustness checks, this may suggest that the sample size is adequate for preliminary evaluation. In many research studies, smaller datasets effectively establish proof of concept before scaling up to larger datasets for further validation.

TABLE II PERFORMANCE RESULTS OF RANDOM FOREST, NAÏVE BAYES, AND K-NN ALGORITHMS, AFTER APPLYING BAGGING METHOD AND OPTIMIZED BY PSO

Algorithms	**Accuracy (%)**	**Precision (%)**	***Recall* (%)**	**AUC**
Random Forest + Bagging + PSO	99.25	98.12	100.00	0.999
k-NN + *Bagging* + PSO	94.50	92.87	93.3	0.973
Naïve Bayes + *Bagging* + PSO	97.25	93.73	100	0.995
The XGBoost model [39]	95	97	98	97
SVM model [40]	91	N/A	N/A	96

Table II presents the performance results of the Random Forest, Naïve Bayes, and k-NN algorithms after applying the Bagging method and optimizing them with PSO. Each algorithm is enhanced using Bagging and PSO techniques. The Random Forest with Bagging and PSO delivers the best performance, achieving 99.25% accuracy, 98.12% precision, 100.00% recall, and an AUC of 0.999. The k-NN algorithm with Bagging and PSO attains 94.50% accuracy, 92.87% precision, 93.33% recall, and an AUC of 0.973. Meanwhile, Naïve Bayes with Bagging and PSO records 97.25% accuracy, 93.73% precision, 100.00% recall, and an AUC of 0.995. Thus, Random Forest with Bagging and PSO demonstrates the best performance across accuracy, precision, recall, and AUC, followed by Naïve Bayes with Bagging and PSO, and k-NN with Bagging and PSO. In addition to using matrices to evaluate the performance of this experiment, the ROC-AUC curve can also be utilized. The comparison of ROC-AUC curves between the Random Forest, Naive Bayes, and k-NN algorithms using the Bagging method optimized with PSO is shown in Fig. 3, 4, 5 and 6.

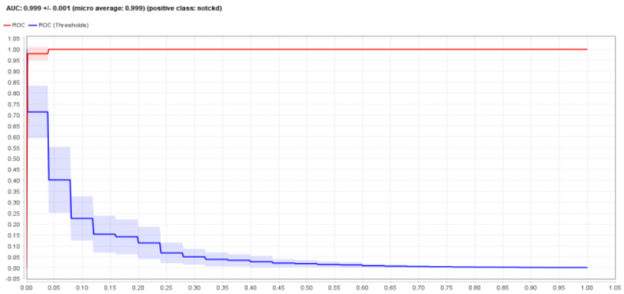


Fig. 3. The experimental results of the ROC-AUC curve for the Random Forest algorithm using the Bagging method optimized with PSO.

The performance of this algorithm in identifying CKD is highly satisfactory. As shown in Fig. 3, the algorithm achieves an Area Under the Curve (AUC) of 0.999, which falls under the category of Excellent Classification.

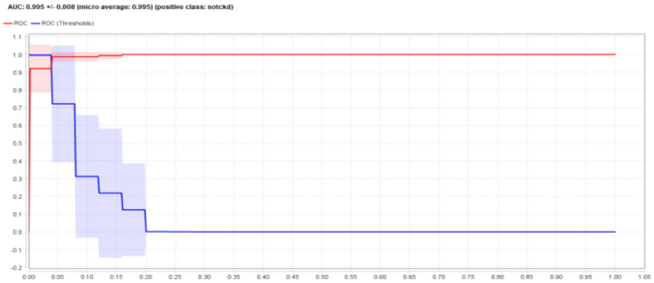


Fig. 4. Experimental results of the ROC with AUC of 0.995 ± 0.008 curve for the Naïve Bayes algorithm using the PSO-based bagging method.

The algorithm performs exceptionally well in identifying Chronic Kidney Disease (CKD). As shown in Fig. 4, it achieves an Area Under the Curve (AUC) of 0.995, which is classified as "Excellent Classification." Additionally, the algorithm maintains strong performance with an AUC of 0.973, also falling under the "Excellent Classification" category.

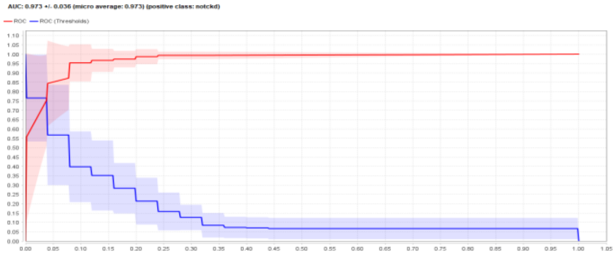


Fig. 5. Experimental results of the ROC with AUC of 0.873 ± 0.014 curve for the Naïve Bayes algorithm using the bagging method optimized with PSO.

Fig. 5 presents a comparison of the feature weights generated by the Random Forest, Naïve Bayes, and k-NN algorithms, utilizing the Bagging method optimized with Particle Swarm Optimization (PSO). It illustrates the experimental results of feature weights for the Random Forest algorithm when employing the Bagging method optimized with PSO. Whilst Fig. 6 displays the attributes of the Random Forest algorithm using the Bagging method optimized with PSO. The figure highlights 24 attributes, each accompanied by its corresponding weight.

Moreover, the graph displays the performance of a binary classification model using the ROC (Receiver Operating Characteristic) and PRC (Precision-Recall Curve). With an ROC AUC of 0.873, the model effectively distinguishes between positive and negative classes, while the PRC AUC of 0.913 highlights its strong performance in imbalanced datasets, particularly where the positive class is prioritized. The ROC curve (red line) shows the trade-off between the true positive rate and the false positive rate, with a steep increase early on, indicating strong performance at low false positive rates. Likewise, the PRC curve (blue line) focuses on the balance between precision and recall, demonstrating high precision even at higher recall levels, which is critical when false positives are costly. The narrow confidence bands at the start of both curves suggest consistent performance across thresholds. Overall, the model exhibits strong classification performance with high AUC values, making it well-suited for tasks requiring precise identification of positive instances, especially in imbalanced datasets.

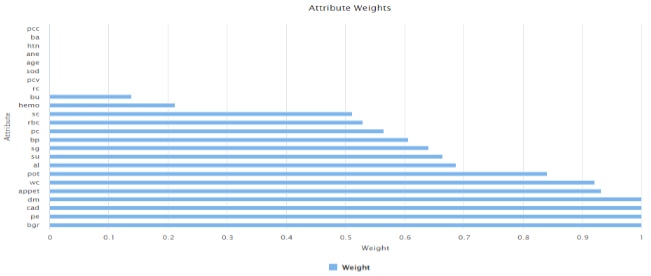


Fig. 6. Visualization of features generated by the Random Forest algorithm using the bagging method optimized with PSO.

Fig. 6 displays the relative importance of various attributes within a dataset. The length of each bar represents the weight assigned to the corresponding attribute, indicating its influence or significance in the analysis. Attributes with longer bars, such as "cad", "sgot", and "alk", are deemed more crucial than those with shorter bars like "pc", "hba", and "alb." This visualization likely aids in feature selection for machine learning models, factor analysis to explain variance, or risk assessment to identify high-risk factors.

The application of the Random Forest algorithm with the Bagging method optimized with PSO results in feature weights r the 24 attributes used, as shown in Fig. 5 and 6. The weights are as follows: rbc 0.529, pc 0.565, dm 1, cad 1, appet 0.932, pe 1, bp 0.606, sg 0.640, al 0.687, su 0.664, bgr 1, bu 0.138, sc 0.511, pot 0.841, hemo 0.212, wc 0.921.

Fig. 6 visualizes the features generated by the Naïve Bayes algorithm using the Bagging method optimized with Particle Swarm Optimization (PSO). The application of the Naïve Bayes algorithm combined with the Bagging method, enhanced by PSO, produces feature weights for the 24 attributes, as illustrated also in Fig. 6. The weights are as follows: rbc (0.950), pc (0.981), pcc (1), dm (1), pe (0.414), age (1), bp (1), sg (0.831), al (0.531), bgr (0.323), bu (1), sc (0.467), sod (1), pot (0.489), hemo (0.826), pcv (1), wc (1), and rc (1). The application of the k-NN algorithm with the Bagging method optimized with PSO results in feature weights for the 24 attributes used, as shown in Fig.6. The weights are as follows: pc 0.725, pcc 0.986, ba 1, htn 1, dm 1, appet 1, ane 1, bp 0.165, sg 1, su 0.642, sc 1, sod 0.139, pot 1, hemo 1, pcv 0.360.

TABLE III COMPARISON OF FEATURE WEIGHTS BETWEEN RANDOM FOREST, NAÏVE BAYES, AND K-NN ALGORITHMS USING THE BAGGING METHOD OPTIMIZED WITH PSO

Attribute	**k-NN + BG + PSO**	**Naïve Bayes + BG + PSO**	**Random Forest + BG + PSO**
Albumin (al)	0	0.531	0.687
Sugar (su)	0.642	0	0.664
Red Blood Cells (rbc)	0	0.950	0.529
Pus Cell (pc)	0.725	0.981	0.565
Pus Cell clumps (pcc)	0.986	1	0
Bacteria (ba)	1	0	0
Blood Glucose Random (bgr)	0	0.323	1
Blood Urea (bu)	0	1	0
Serum Creatinine (sc)	1	0.467	0.138
Sodium (sod)	0.139	1	0
Potassium (pot)	1	0.489	0.841
Haemoglobin (hemo)	1	0.826	0.212

Packed Cell Volume (pcv)	0.360	1	0
White Blood Cell Count (wc)	0	1	0.921
Red Blood Cell Count (rc)	0	1	0
Hypertension (htn)	1	0	0
Diabetes Mellitus (dm)	1	1	1
Coronary Artery Disease (cad)	0	0	1
Appetite (appet)	1	0	0.932
Pedal Edema (pe)	0	0.414	1
Anaemia (ane)	1	0	0
Albumin (al)	0	0.531	0.687
Sugar (su)	0.642	0	0.664
Red Blood Cells (rbc)	0	0.950	0.529
Pus Cell (pc)	0.725	0.981	0.565
Pus Cell clumps (pcc)	0.986	1	0
Bacteria (ba)	1	0	0
Blood Glucose Random (bgr)	0	0.323	1
Blood Urea (bu)	0	1	0

Based on the weighting results, the feature weights for the three algorithms (k-NN, Naïve Bayes, and Random Forest) using the Bagging method optimized with PSO are shown. The Random Forest algorithm produces a weight combination that enhances model performance in classification. Note that several attributes (e.g., rbc 0.529, pc 0.565, dm 1, cad 1, appet 0.932, pe 1, bp 0.606, sg 0.640, al 0.687, su 0.664, bgr 1, bu 0.138, sc 0.511, pot 0.841, hemo 0.212, wc 0.921) in the Random Forest feature weights are close to or equal to 1, indicating their significant influence on classification. Moreover, attributes with significant weights can provide valuable information for the classification model. The Random Forest algorithm improves accuracy, precision, and recall by finding the optimal weight combinations for relevant attributes using the Bagging method optimized with PSO. Additionally, some attributes with a weight of 0 are automatically discarded as they have no impact on the process. Thus, the feature weighting in the Random Forest algorithm using the Bagging method optimized with PSO proves to be superior in this case.

In the evaluation phase of the research, a comparison of experimental results was conducted using three classification algorithms (Random Forest, Naïve Bayes, and k-NN) with the Bagging method optimized with Particle Swarm Optimization (PSO). The results show a significant difference when using PSO feature selection. Experiments without feature selection showed the highest accuracy for Random Forest (98.75%), followed by Naïve Bayes (94.75%) and k-NN (73.75%). After optimization with PSO and using the Bagging method, accuracy improved for all algorithms. Random Forest achieved the highest accuracy (99.25%) with a precision of 98.12%, recall of 100.00%, AUC of 0.999, and 16 features influencing the score. The high accuracy value is influenced by several factors, including parameters; the setting of parameters in the model affects accuracy. If the parameters used are not suitable for the data or cannot predict accurately, the accuracy value will decrease. The performance of the AUC [35] is classified into five categories, as shown in Table IV.

TABLE IV CLASSIFICATION CATEGORIES BASED ON AUC VALUE

AUC Value	Classification Category
0.90 - 1.00	Excellent
0.80 - 0.90	Good
0.70 - 0.80	Fair
0.60 - 0.70	Poor
0.50 - 0.60	Fail

According to the AUC classification table, the Random Forest algorithm falls into the "Excellent" category with an AUC value of 0.999 and generates 15 feature weights, each with a corresponding value. This indicates that the Random Forest algorithm is highly effective for analysis. Based on the above classification, it can be concluded that the Random Forest algorithm optimized with Particle Swarm Optimization (PSO) and using the Bagging method is a Very Good algorithm and suitable for analysis.

As describe on Table IV that Receiver Operating Characteristic (ROC) curve and its corresponding Area Under the Curve (AUC) value provide a quantitative measure of a model's classification performance. According to Table IV, which categorizes classification performance based on AUC values, the first model, with an AUC of 0.995 ± 0.008, falls into the "Excellent" category (0.90 - 1.00). This indicates that the model is highly effective at distinguishing between the positive (notckd) and negative (ckd) classes, with minimal misclassification. The near-perfect AUC score suggests high sensitivity and specificity, making it a highly reliable classification tool.

In comparison, the second model, with an AUC of 0.873 ± 0.014, falls into the "Good" category (0.80 - 0.90). While still strong, this AUC value reflects a slightly lower ability to differentiate between classes compared to the first model. The confidence intervals indicate some variability in performance, but the model remains effective for classification purposes. Overall, the first model demonstrates exceptional classification ability, making it particularly suitable for applications requiring high precision and reliability, such as medical diagnosis. The second model, though slightly less precise, still performs well and could benefit from further optimization through feature selection or model tuning to enhance its performance.

IV. CONCLUSION AND RECOMMENDATION

Optimized with PSO achieved the highest performance The research on predicting chronic kidney disease using the Random Forest, Naïve Bayes, and k-NN algorithms with the Bagging approach optimized with Particle Swarm Optimization (PSO) has been outlined.

The use of the Bagging method with Particle Swarm Optimization (PSO) feature selection improves the accuracy, precision, recall, and AUC values for the Random Forest, Naïve Bayes, and k-NN algorithms. In testing, Random Forest with the Bagging method with an accuracy of 99.25%, precision of 98.12%, recall of 100.00%, and an AUC of 0.999, all falling into the "Excellent" category. This indicates that the model has a high level of agreement between the predictions made by the model and the actual values in the test data. In other words, the higher the AUC value, the better the model is at predicting the class or target variable.

The Bagging approach with Particle Swarm Optimization (PSO) enhances the performance of Random Forest, Naïve Bayes, and k-NN in predicting chronic kidney disease, several limitations must be addressed. The model's high accuracy, precision, recall, and AUC values come from a single dataset, limiting generalizability to diverse populations. Without external validation, its reliability in real-world settings remains uncertain. Additionally, potential biases, such as class imbalances, may affect performance. The study also lacks an assessment of the model's clinical usability and interpretability. Future research should validate the model across diverse datasets, address biases, and ensure practical clinical integration.

Acknowledgment

The authors express their gratitude to the Institute Informatics and Business Darmajaya, Indonesia for the valuable support in this research.

References

[1] R. Alaghehbandan, F. Siadat, and K. Trpkov, "What's new in the WHO 2022 classification of kidney tumours?," 2023. doi: 10.32074/1591-951X-818.

[2] T. A. Berezina, I. M. Fushtey, A. A. Berezin, S. V. Pavlov, and A. E. Berezin, "Predictors of Kidney Function Outcomes and Their Relation to SGLT2 Inhibitor Dapagliflozin in Patients with Type 2 Diabetes Mellitus Who Had Chronic Heart Failure," *Adv Ther*, vol. 41, no. 1, 2024, doi: 10.1007/s12325-023-02683-y.

[3] R. K. Halder *et al.*, "ML-CKDP: Machine learning-based chronic kidney disease prediction with smart web application," *J Pathol Inform*, vol. 15, 2024, doi: 10.1016/j.jpi.2024.100371.

[4] C. D. Priyanka, S. J. S. Keerthana, M. R. Babu, and B. S. K. Devi, "IoT based prediction of chronic kidney disease," in *AIP Conference Proceedings*, 2024. doi: 10.1063/5.0190642.

[5] K. S. Suthar *et al.*, "Urinary Screening for Early Detection of Kidney Diseases," *Indian J Pediatr*, vol. 85, no. 8, 2018, doi: 10.1007/s12098-017-2494-y.

[6] H. Ilyas *et al.*, "Chronic kidney disease diagnosis using decision tree algorithms," *BMC Nephrol*, vol. 22, no. 1, 2021, doi: 10.1186/s12882-021-02474-z.

[7] S. Malakar, S. Sen, S. Romanov, D. Kaplun, and R. Sarkar, "Role of transfer functions in PSO to select diagnostic attributes for chronic disease prediction: An experimental study," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 9, 2023, doi: 10.1016/j.jksuci.2023.101757.

[8] S. Kavi Priya and N. Saranya, "An Intelligent Approach for Accurate Prediction of Chronic Diseases," *Computer Systems Science and Engineering*, vol. 46, no. 2, 2023, doi: 10.32604/csse.2023.031761.

[9] R. H. Aswathy *et al.*, "Optimized tuned deep learning model for chronic kidney disease classification," *Computers, Materials and Continua*, vol. 70, no. 2, 2022, doi: 10.32604/cmc.2022.019790.

[10] A. Zizaan and A. Idri, "Evaluating and comparing bagging and boosting of hybrid learning for breast cancer screening," *Sci Afr*, vol. 23, 2024, doi: 10.1016/j.sciaf. 2023.e01989.

[11] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Particle Swarm Optimization or Differential Evolution—A comparison," *Eng Appl Artif Intell*, vol. 121, 2023, doi: 10.1016/j.engappai.2023.106008.

[12] V. KR, M. S. Maharajan, B. K, and N. Sivakumar, "Classification of adaptive back propagation neural network along with fuzzy logic in chronic kidney disease," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 7, 2024, doi: 10.1016/j.prime.2024.100463.

[13] K. C. Leung, W. W. S. Ng, Y. P. Siu, A. K. C. Hau, and H. K. Lee, "Deep learning algorithms for predicting renal replacement therapy initiation in CKD patients: a retrospective cohort study," *BMC Nephrol*, vol. 25, no. 1, 2024, doi: 10.1186/s12882-024-03538-6.

[14] L. Qadrini, "Undersampling dan K-Fold Random Forest Untuk Klasifikasi Kelas Tidak Seimbang," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 4, 2023, doi: 10.47065/bits.v4i4.3141.

[15] A. M. Adeshina, "Prediction of Diabetes Mellitus using Machine Learning Algorithms: Comparative Analysis of K-Nearest Neighbor, Random Forest and Logistic Regression," *SLU Journal of Science and Technology*, 2023, doi: 10.56471/slujst.v6i.319.

[16] A. K. Chaudhuri, D. Sinha, D. K. Banerjee, and A. Das, "A novel enhanced decision tree model for detecting chronic kidney disease," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 10, no. 1, 2021, doi: 10.1007/s13721-021-00302-w.

[17] Muhasshanah, M. Tohir, D. A. Ningsih, N. Y. Susanti, A. Umiyah, and L. Fitria, "Comparison of the performance results of c4.5 and random forest algorithm in data mining to predict childbirth process," *CommIT Journal*, vol. 17, no. 1, 2023, doi: 10.21512/commit.v17i1.8236.

[18] H. fen Chen *et al.*, "Lipid parameters, adipose tissue distribution and prognosis prediction in chronic kidney Disease patients," *Lipids Health Dis*, vol. 23, no. 1, 2024, doi: 10.1186/s12944-024-02004-4.

[19] S. Kavi Priya and N. Saranya, "An Effective Chronic Disease Prediction using Multi-Objective Firefly Optimisation Random Forest Algorithm," *IETE J Res*, vol. 70, no. 1, 2024, doi: 10.1080/03772063.2022.2108916.

[20] J. Guo, C. Chen, H. Wen, G. Cai, and Y. Liu, "Prediction model of goaf coal temperature based on PSO-GRU deep neural network," *Case Studies in Thermal Engineering*, vol. 53, 2024, doi: 10.1016/j.csite.2023.103813.

[21] E. A. Aner, M. I. Awad, and O. M. Shehata, "Performance evaluation of PSO-PID and PSO-FLC for continuum robot's developed modeling and control," *Sci Rep*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-023-50551-0.

[22] X. You *et al.*, "A PSO-CNN-Based Deep Learning Model for Predicting Forest Fire Risk on a National Scale," *Forests*, vol. 15, no. 1, 2024, doi: 10.3390/f15010086.

[23] K. ADEM, "Diagnosis of Chronic Kidney Disease using Random Subspace Method with Particle Swarm Optimization," *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi*, vol. 10, no. 3, 2018, doi: 10.29137/umagd.472881.

[24] J. Gao, Z. Wang, T. Jin, J. Cheng, Z. Lei, and S. Gao, "Information gain ratio-based subfeature grouping empowers particle swarm optimization for feature selection," *Knowl Based Syst*, vol. 286, 2024, doi: 10.1016/j.knosys.2024.111380.

[25] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network Intrusion Detection Based on PSO-Xgboost Model," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2982418.

[26] M. de Rooij and W. Weeda, "Cross-Validation: A Method Every Psychologist Should Know," *Adv Methods Pract Psychol Sci*, vol. 3, no. 2, 2020, doi: 10.1177/2515245919898466.

[27] S. Y. Irianto, R. Yunandar, M. S. Hasibuan, D. A. Dewi, and N. Pitsachart, "Early Identification of Skin Cancer Using Region Growing Technique and a Deep Learning Algorithm," *HighTech and Innovation Journal*, vol. 5, no. 3, pp. 640–662, Sep. 2024, doi: 10.28991/HIJ-2024-05-03-07.

[28] J. Lei, "Cross-Validation With Confidence," *J Am Stat Assoc*, vol. 115, no. 532, 2020, doi: 10.1080/01621459.2019.1672556.

[29] A. Seraj *et al.*, "Cross-validation," in *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*, 2022. doi: 10.1016/B978-0-12-821285-1.00021-X.

[30] S. Chikkalingaiah, S. A. P. R. H. Prasad, and L. D. Uggregowda, "Classification techniques using gray level co-occurrence matrix features for the detection of lung cancer using computed tomography imaging," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, 2023, doi: 10.11591/ijece.v13i5.pp5135-5146.

[31] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf Sci (N Y)*, vol. 507, 2020, doi: 10.1016/j.ins.2019.06.064.

[32] A. H. Sayed, "Bagging and Boosting," in *Inference and Learning from*

Data, 2023. doi: 10.1017/9781009218276.014.

[33] W. Zhai *et al.*, “A Bagging-SVM field-road trajectory classification model based on feature enhancement,” *Comput Electron Agric*, vol. 217, 2024, doi: 10.1016/j.compag.2024.108635.

[34] V. Sülar and Y. Seki, “A review on fabric bagging: the concept and measurement methods,” 2018. doi: 10.1080/00405000.2017.1354450.

[35] P. Bühlmann and B. Yu, “Analyzing bagging,” 2002. doi: 10.1214/aos/1031689014.

[36] N. Alturki *et al.*, “Improving Prediction of Chronic Kidney Disease Using KNN Imputed SMOTE Features and TrioNet Model,” *CMES - Computer Modeling in Engineering and Sciences*, vol. 139, no. 3, 2024, doi: 10.32604/cmes.2023.045868.

[37] Q. A. A’yuniyah and M. Reza, “Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Jurusan Siswa Di Sma Negeri 15 Pekanbaru,” *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, vol. 3, no. 1, 2023, doi: 10.57152/ijirse.v3i1.484.

[38] F. S. N. Khamidah, D. P. Hapsari, and H. Nugroho, “Implementasi Fuzzy Decision Tree Untuk Prediksi Gagal Ginjal Kronis,” *Integer: Journal of Information Technology*, vol. 3, no. 1, 2018, doi: 10.31284/j.integer.2018.v3i1.155.

[39] Z. Chen, Y. Wang, M. T. C. Ying, and Z. Su, “Interpretable machine learning model integrating clinical and elastosonographic features to detect renal fibrosis in Asian patients with chronic kidney disease,” *J Nephrol*, vol. 37, no. 4, 2024, doi: 10.1007/s40620-023-01878-4.

[40] C. Hua *et al.*, “Value of multiparametric magnetic resonance imaging for evaluating chronic kidney disease and renal fibrosis,” *Eur Radiol*, vol. 33, no. 8, 2023, doi: 10.1007/s00330-023-09674-1.