Scientific Journal of Informatics
Vol. 12, No. 1, Feb 2025
p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Integrating C4.5 and K-Nearest Neighbor Imputation with Relief Feature Selection for Enhancing Breast Cancer Diagnosis

Aji Purwinarko[1*], Kholiq Budiman[2], Arif Widiyatmoko[3], Fitri Arum Sasi[4], Wahyu Hardyanto[5]
[1, 2]Information Technology Studies Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia
[3]Science Education Studies Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia
[4]Biology Studies Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia
[5]Physics Studies Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: Breast cancer remains a significant cause of mortality among women, requiring accurate diagnostic methods. Traditional classification models often face accuracy challenges due to missing values and irrelevant features. This investigation advances the classification of breast cancer through the amalgamation of the C4.5 algorithm with K-Nearest Neighbor (KNN) imputation and Relief feature selection methodologies, thereby augmenting data integrity and enhancing classification efficacy.
Methods: The Wisconsin Breast Cancer Database (WBCD) was the core reference for evaluating the proposed methodology. KNN imputation addressed missing values, while Relief selected the most relevant features. The C4.5 algorithm executed training by utilizing data segregations in the corresponding proportions of 70:30, 80:20, and 90:10, with its efficiency gauged through a range of metrics, particularly accuracy, precision, recall, and F1-score.
Result: This innovative methodology achieved the highest classification accuracy of 98.57%, surpassing several existing models. Particularly noteworthy, the strategy being analyzed exhibited remarkable success relative to PSO-C4.5 (96.49%), EBL-RBFNN (98.40%), Gaussian Naïve Bayes (97.50%), and t-SNE (98.20%), demonstrating associated advancements of 2.08%, 0.17%, 1.07%, and 0.37%. These results confirm its effectiveness in handling missing values and selecting relevant features.
Novelty: Unlike prior studies that addressed missing values and feature selection separately, this research integrates both techniques, enhancing classification accuracy and computational efficiency. The findings suggest that this approach provides a reliable breast cancer diagnosis method. Future work could explore deep learning integration and validation on larger datasets to improve generalizability.

Keywords: Breast cancer classification, C4.5, KNN imputation, Relief feature selection, Machine learning
Received February 2025 / **Revised** March 2025 / **Accepted** April 2025

INTRODUCTION

In recent decades, the understanding of breast cancer has changed dramatically due to extensive research into its molecular characteristics [1]. Breast cancer is the most common cancer in women and the leading cause of cancer death among women [2] This condition is caused by the development of abnormal cells in the breast, where the type and characteristics of the malignant cells involved determine the classification of breast cancer [3]. The prevalence of breast cancer is increasing in developing countries due to lifestyle changes, including diet, delayed pregnancy, decreased fertility, and shorter breastfeeding periods [4]. Although survival rates are high when detected early, many women face social, economic, and geographic barriers to accessing timely and affordable breast health services [5]. Manual diagnosis of breast cancer is time-consuming and requires specialized expertise, making machine-based prediction essential to prevent further spread [6]. Recent research using deep learning techniques as part of machine learning (ML) shows great potential in early detection [7] and treatment of breast cancer [8].

ML constitutes a subset of artificial intelligence (AI) methodologies that leverage algorithmic frameworks to enhance efficacy in designated undertakings, including but not limited to classification and prediction, by assimilating insights derived from data devoid of direct instructional guidance. The hallmark of ML is

*Corresponding author.
Email addresses: aji.purwinarko@mail.unnes.ac.id (Purwinarko)*, kholiq.budiman@mail.unnes.ac.id (Budiman), arif.widiyatmoko@mail.unnes.ac.id (Widiyatmoko), tredeef@mail.unnes.ac.id (Sasi), hardy@mail.unnes.ac.id (Hardyanto)
DOI: 10.15294/sji.v12i1.21673

the use of algorithms that learn from data without explicit instructions on how to achieve it [9]. Classification is an important and essential operation in data science and ML [10]. Classification is a supervised ML problem, which focuses on accurately assigning observations to a particular class, either binary with two possible classes or multi-class [11]. In addition to providing accurate results, classification offers insight into how the model makes decisions [6]. The classification process requires labeled data for its development, while most of the data is available in unlabeled form, so it needs to be addressed. According to [12], classification is an essential ML technique that groups data based on the class labels provided. Data labeling is important because well-labeled data will produce a good classification model. The data labeling process must be followed carefully and consistently, as classification requires well-labeled and accountable data [11]. Multi-label classification represents an advancement of the multi-class classification problem, in which a collection of class labels is correlated with a singular instance concurrently [13]. Decision trees are one of the most influential classification techniques in data mining; they have become popular due to their ease of use and visualization for various types of data sets [14].

Decision trees can predict the future by building a primary and representative classification or regression model in the form of a tree structure [15]. This technique is used to classify observations based on their characteristics and make numerical or categorical predictions [16]. The decision tree approach transforms facts into data, valid for exploring data in a tree structure that provides easy-to-understand rules for identifying hidden relationships between input and target variables [17]. Decision tree classification uses pre-assigned labels to ascertain or predict the class of a future data set where class labeling is uncertain [18]. The performance of a decision tree is evaluated based on the accuracy of predicting unobserved events [19]. The three predominant algorithms employed in the context of decision trees include C4.5, Classification and Regression Trees (CART), and Random Forest (RF) [20]. The C4.5 algorithm, the successor to Iterative Dichotomiser 3 (ID3), can handle datasets with different numeric features [21]. C4.5 also supports handling categorical features and missing values and introduces the concept of gain ratio to overcome the bias of ID3 towards features with many different values [22]. One way to improve the accuracy of the classification results is to perform data preprocessing [20]. Handling missing values is often required in the preprocessing phase of a dataset for training and testing a model [23].

Missing values can occur due to various factors, such as missing completely, randomly, or non-randomly, caused by system failures or human errors during data collection or preprocessing. Ignoring or eliminating missing values can lead to biased or misinformed analysis, so it is essential to handle them before analyzing the data [24]. Missing values can also reduce the study's statistical power or lead to inaccurate estimates and conclusions. Therefore, although classification algorithms are used to handle numerical features, data normalization and handling missing values are considered significant issues in the data preprocessing stage [25]. Some methods for handling missing values include using K-Nearest Neighbor (KNN) [26], replacing with the mean value, CART [27] and Naïve Bayes [28]. The KNN algorithm employs straightforward similarity formulas, such as the Euclidean distance, to address classification issues. KNN offers the advantage of providing recommendations with high accuracy and relatively fast. KNN method generally uses two basic formulas to measure the similarity between training and test data: Euclidean distance and cosine similarity [28]. Notwithstanding its apparent straightforwardness, the KNN algorithm has demonstrated considerable efficacy. This method is non-parametric and instance-based, suitable for lazy learning, and, in the context of classification, is used to determine the class of newly discovered unlabeled objects [29].

In addition, in ML, one of the crucial preprocessing stages is feature selection, which aims to minimize the dimensionality of the dataset by selecting the most significant features from the original set [30]. This feature selection allows the identification and prioritization of critical features, thereby increasing the efficiency of the ML algorithm by focusing on relevant data [31]. There exists a multitude of algorithms that may be utilized for feature selection, including RF [32], linearity-based methods [33], improved butterfly optimization algorithm (IBOA), particle swarm optimization (PSO) [30], genetic algorithms (GA), principal component analysis, chi-square, and relief [34]. According to [35], Relief is one of the popular feature selection algorithms. Relief and its variations have shown high effectiveness in practice, evaluating features based on their ability to distinguish closely related instances [36].

Numerous prior investigations have concentrated on the utilization of the C4.5 algorithm in the context of breast cancer diagnosis. However, there are still obstacles in terms of accuracy due to missing values in the data and less than optimal feature selection. Several researchers have used various imputation and feature

selection methods, but studies that combine the KNN method for imputation with Relief as feature selection are still limited. Accordingly, this academic investigation aspires to refine the effectiveness of the C4.5 algorithm in the realm of breast cancer diagnosis. This research intends to enhance the C4.5 algorithm by integrating a KNN-based missing value imputation technique and applying the Relief method for feature selection, aiming to yield a more precise predictive model for breast cancer.

METHODS

Dataset

This study uses the Wisconsin Breast Cancer Database (WBCD) from the UCI Repository of Machine Learning [37]; it was obtained from the University of Wisconsin Hospital, Madison, by Dr. William H. Wolberg. The characteristics inherent to the dataset are delineated in Table 1. The WBCD dataset comprises 699 records of 11 features, with ten numeric features and one categorical feature.

Table 1. WBCD dataset features

Variable Name	Role	Type	Description	Missing Values
Sample_code_number	ID	Categorical	1-10	no
Clump_thickness	Feature	Integer	1-10	no
Uniformity_of_cell_size	Feature	Integer	1-10	no
Uniformity_of_cell_shape	Feature	Integer	1-10	no
Marginal_adhesion	Feature	Integer	1-10	no
Single_epithelial_cell_size	Feature	Integer	1-10	no
Bare_nuclei	Feature	Integer	1-10	yes
Bland_chromatin	Feature	Integer	1-10	no
Normal_nucleoli	Feature	Integer	1-10	no
Mitoses	Feature	Integer	1-10	no
Class	Targer	Binary	2 = benign, 4 = malignant	no

Research stages

This research begins with data collection, followed by preprocessing involving two main steps: missing value imputation with KNN and feature selection stage using relief algorithm. The following procedure consists of the division of the dataset into training and testing subsets by the specified proportions of 70:30, 80:20, and 90:10. The ensuing procedure is designed to ascertain that, if the data constitutes training data, such data shall be employed by the C4.5 algorithm for the construction of a predictive model and subsequently subjected to evaluation utilizing testing data. After evaluation, the model accuracy is calculated using the confusion matrix and ends with reporting the model accuracy. Overall, Figure 1 shows the research method.

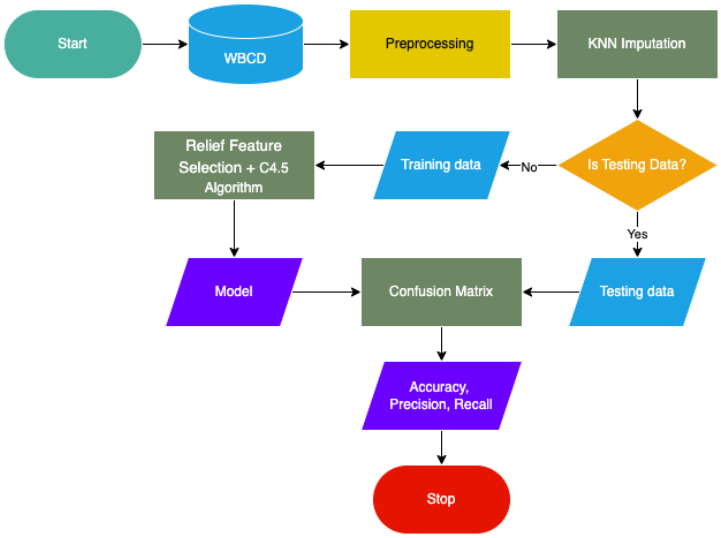

Figure 1. Research method flowchart

KNN

KNN is a well-known and widely used classification algorithm that works on finding the k nearest neighbors of a query point [38]. Determining the value of k is crucial because it affects the number of

neighbors considered and significantly impacts the model's efficiency [39]. The performance of KNN is highly dependent on the accuracy of the selection of the k parameter, which is usually determined through cross-validation due to its dependence on the training dataset [40]. The algorithm's performance is affected by variations in the value of k and the high variance in the training dataset, with prediction accuracy tending to decrease as k approaches larger values [41]. Various distance metrics have been applied in KNN, including Euclidean, Minkowski, Manhattan, and Canberra distances. The choice of distance metric has a significant impact on the performance of KNN, with modifications such as the Modified Euclidean-Canberra Blend Distance (MECBD) metric shown to improve class prediction efficiency [42]. The KNN algorithm is described in Figure 2.

In classification, KNN assigns the majority class label of the nearest neighbors to an unknown object, while in regression, it predicts the value of an unseen data point as a linear combination of its k neighbors on a graph [39]. This algorithm's simplicity and non-parametric nature allow it to adapt to complex models without assumptions about the data distribution, making it a popular choice for simple classification models [43]. Its effectiveness, ease of implementation, and ability to add new data to the training set at any time make KNN a widely used classification algorithm [40].

Algorithm 1 K-Nearest Neighbors (KNN) Algorithm

Require: Training data X_{train}, Test data X_{test}, Number of neighbors K
Ensure: Predicted class or value $\hat{y}$
for each test instance x_{test} in X_{test} **do**
  for each training instance x_{train} in X_{train} **do**
    Calculate distance $d(x_{\text{test}}, x_{\text{train}})$ using Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

  end for
  Sort distances in ascending order
  Select the top K neighbors
  if classification task **then**
    Count the frequency of each class among the K neighbors
    Assign the most frequent class to $\hat{y}$ using

$$\hat{y} = \text{mode}(y_i), \quad i = 1, 2, ..., K \quad (2)$$

  else if regression task **then**
    Calculate the mean value of the K neighbors

$$\hat{y} = \frac{1}{K}\sum_{i=1}^{K} y_i \quad (3)$$

  end if
end for

Figure 2. KNN algorithm

Relief feature selection

Feature selection is a technique to reduce datasets' dimensionality and improve ML models' performance and efficiency [44]. Various feature selection algorithms have been developed, including GA, PSO, ensemble methods such as RF and Gradient Boosting Machines, and metaheuristic approaches such as the firefly algorithm and tabu search [45]. These techniques can be grouped into filter, wrapper, and embedded methods, each with its unique approach to selecting relevant features [46]. The benefits of feature selection include improved learning performance, reduced computational cost, and improved model interpretability [47]. This process plays a vital role in improving the quality of data models, leading to more accurate performance predictions and enhancing the quality of education [48]. The presence of irrelevant features can decrease the classification accuracy and increase the training and classification time, making feature selection a crucial step in the machine learning process [49]. An additional algorithm utilized for feature selection is the Relief algorithm.

The Relief algorithm, as indicated by Figure 3, is a feature selection method initially designed for binary classification to identify the two nearest neighbors of the same and different classes, known as nearest hits and nearest hits [36]. This method is used to form a subset of relevant features for effective data classification [50], [51]. Relief evaluates the quality of features individually and selects the features with the highest scores for subset formation [51], [52]. Relief-based algorithms have been applied in various fields, including medical data analysis for lung cancer and heart disease prediction, demonstrating their effectiveness in improving classification [53].

Algorithm 2 Relief Feature Selection Algorithm

Require: Dataset D with instances X, features F, number of iterations m
Ensure: Weight vector W for features

Initialize weight vector $W = \mathbf{0}$
for $i = 1$ to m **do**
  Randomly select an instance R from D
  Find nearest hit H: nearest neighbor of R with the same class
  Find nearest miss M: nearest neighbor of R with a different class
  for each feature $f \in F$ **do**
    Update weight $W[f]$ using:

$$W[f] = W[f] - \frac{1}{m} \cdot \Delta(f, R, H) + \frac{1}{m} \cdot \Delta(f, R, M) \tag{4}$$

  end for
end for
function $\Delta(f, R, N)$
  if f is discrete **then**

$$return \begin{cases} 0, & \text{if } f(R) = f(N) \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

  else

$$Normalize\ difference: \Delta(f, R, N) = |f(R) - f(N)|/(f_{\max} - f_{\min}) \tag{6}$$

  end if
end function
return W

Figure 3. Relief feature selection algorithm

C4.5

The C4.5 algorithm, as shown by Figure 4, is a development of ID3 that introduces the concept of information gain level, making it superior in certain aspects [54]. This algorithm is known for its ease of understanding and high accuracy in classification tasks, making it a widely used method in various fields, including medical diagnosis [21]. C4.5 can handle datasets with different numeric attributes. It is often optimized using cross-validation to determine the tree's depth and the leaf nodes' minimum size, with a post-training pruning strategy to improve its performance [55]. However, in some cases, this algorithm can produce less than optimal accuracy, especially when dealing with many classes, which can increase decision-making time [56]. As one of the classic Shannon entropy-based decision trees, C4.5 uses the gain ratio to determine attribute separation, making it the basis for further method development as shown by Equation (7) and Equation (8) [57]. In addition, in a federative learning environment, decision tree structures such as C4.5, ID3, and CART require special techniques to combine decision paths without causing bias or overfitting while maintaining the robustness and generalizability of the resulting model [58].

Algorithm 3 C4.5 Decision Tree Algorithm

Require: Dataset S, Features F
Ensure: Decision Tree T

function BUILDTREE(S, F)
  if S is pure or F is empty **then**
    return leaf node with majority class
  end if
  for each feature A in F **do**
    Calculate $Entropy(S)$

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{7}$$

    Calculate $Gain(S, A)$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{8}$$

    Calculate $SplitInfo(A)$

$$SplitInfo(A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|} \tag{9}$$

    Calculate $SplitInfo(A)$

$$GainRatio(A) = \frac{Gain(S, A)}{SplitInfo(A)} \tag{10}$$

  end for
  $A_{best} \leftarrow$ feature with highest $GainRatio$
  Create a decision node N for A_{best}
  for each value v of A_{best} **do**
    $S_v \leftarrow$ subset of S where $A_{best} = v$
    if S_v is empty **then**
      Add a leaf node with majority class of S
    else
      $N_v \leftarrow$ BUILDTREE(S_v, $F - \{A_{best}\}$)
      Add N_v as a child of N
    end if
  end for
  return N
end function
$T \leftarrow$ BUILDTREE(S, F)
Perform pruning on T
return T

Figure 4. C4.5 decision tree algorithm

Confusion matrix

A confusion matrix constitutes a quantitative framework employed to examine the distribution of inaccuracies within the classification process by juxtaposing the actual classes with the predicted classes. This matrix is represented in a tabular format, where the rows denote the actual classes, while the columns signify the predicted classes, thereby facilitating the visualization of the performance metrics of the machine learning model [59]. Typically, a confusion matrix is made up of four key components: true positive (TP), true negative (TN), false positive (FP), and false negative (FN), which play a crucial role in determining various evaluative metrics such as true positive rate (TPR), true negative rate (TNR), false positive rate, and false negative rate [60]. For instance, a confusion matrix characterized by dimensions of 2x2 is illustrated in Table 2.

Table 2. Confusion matrix

		True/Actual Class	
		Positive (P)	**Negative (N)**
Predicted Class	**Positive (P)**	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

The confusion matrix offers a comprehensive analysis of the performance metrics of the model, which encompasses:

1. Accuracy serves as an indicator of the frequency with which a classification model produces correct predictions.

$$\text{Accuracy: } (TP + TN) / (TP + TN + FP + FN) \quad (11)$$

2. Precision measures how accurately the model predicts the positive class.

$$\text{Precision: } TP / (TP + FP) \quad (12)$$

3. Recall, or sensitivity, measures how well the model can detect all positive cases.

$$\text{Recall: } TP / (TP + FN) \quad (13)$$

4. F1-Score is the harmonic mean of precision and recall. It establishes an equilibrium between precision and recall, which is particularly advantageous when a disparity exists between the positive and negative classifications.

$$\text{F1-Score: } 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (14)$$

RESULTS AND DISCUSSIONS

Preprocessing

The following process removes the sample_cod_number feature from the dataset. The sample_cod_number feature is removed from the dataset because it is considered irrelevant and is a data sequence number. So, after removing the feature, there will be 10 selected features from the 11 existing features.

KNN imputation

Table 4.2 shows the results of checking for missing values in the dataset. In Table 4.2, the "Bare Nuclei" column has 16 missing values, equivalent to 2.32% of the total data. Concurrently, the remaining columns exhibit the absence of any missing values. Identifying columns with missing values allows us to take appropriate action. Handling is done using KNN Imputer, which fills the empty columns with KNN values obtained from the nearest neighbors. KNN works by calculating weight mean estimation based on k, where k is the number of closest observations to be used. The k used is 10, and then the Euclidean distance determines the k nearest neighbors for a particular test sample. The resulting prediction depends directly on the specific set of neighbors chosen.

Relief feature selection and C4.5

Feature selection in this study aims to reduce redundancy and increase model efficiency by retaining features with a high correlation with the target (Class) and removing features with a high correlation with each other unless both provide important unique information. A comprehensive analysis was conducted utilizing a correlation heat map to ascertain the interrelationships among features within the WBCD dataset, as illustrated in Figure 2. This diagram delineates the Pearson correlation coefficient pertaining to the various attributes present in the dataset, with the values fluctuating between -1 and 1, wherein values approaching 1 signify a significant positive correlation; in contrast, values approximating -1 imply a substantial negative correlation.

The analysis results found that the features Uniformity of Cell Size and Uniformity of Cell Shape had a very high correlation (0.91), indicating that both provide similar information, so one can be considered for

removal in the feature selection process. In addition, the Clump Thickness feature also showed a relatively high correlation with Class (0.72), indicating its relevance to class prediction in the classification model. In contrast, the feature denoted as Mitoses exhibits a diminished correlation with Class (0.42), suggesting that its influence on the predictive model is less significant than the other features.

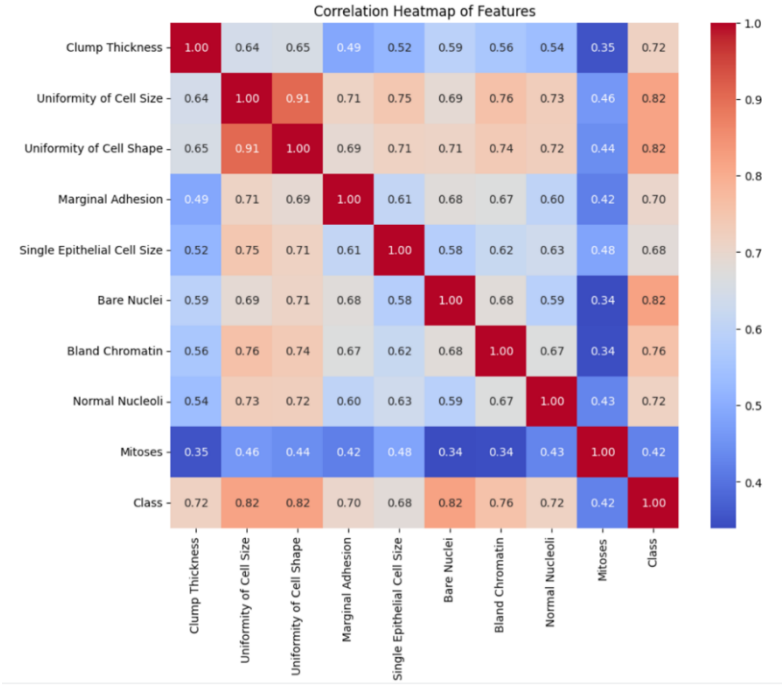

Figure 2. Correlation Heatmap of Features

Subsequent feature selection was executed utilizing the Relief algorithm, culminating in identifying the most optimal 6 features from the original 10 features in the dataset. The selected features included Uniformity of Cell Size, Uniformity of Cell Shape, Bare Nuclei, Single Epithelial Cell Size, Bland Chromatin, and Normal Nucleoli. The identification of anomalous data points within the dataset was further examined through the utilization of the Box Plot (Figure 3). This visual illustration signifies that the Bare Nuclei attribute showcases a broader diversity of values when contrasted with the other parameters, while elements like Single Epithelial Cell Size and Bland Chromatin present outlier cases that require thorough evaluation in the data preprocessing procedure. This examination serves as the foundation for ascertaining the most effective approach to enhance the efficacy of the employed classification model.

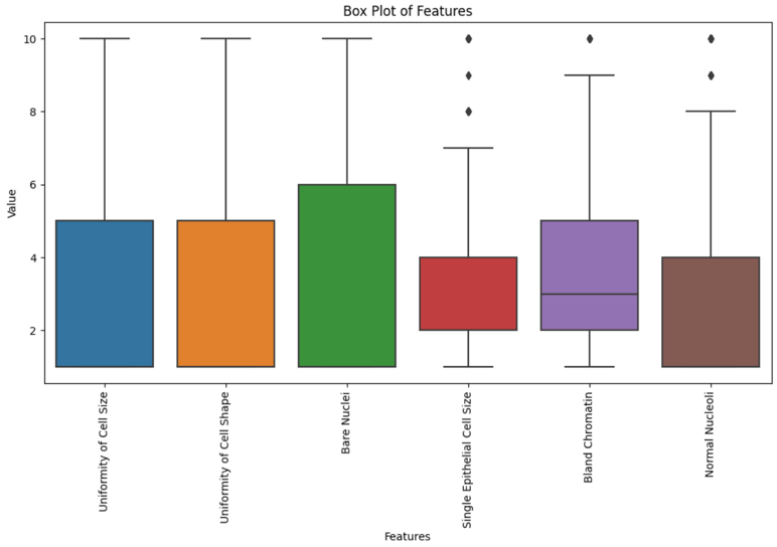

Figure 3. Blox plot of features

The performance evaluation of the classification model with various data splits is shown in Table 3. The empirical study illustrates that shifts in the distribution of training to test data considerably modify the model's accuracy, precision, recall, and F1 score.

Table 3. Experiment results using data splitting

Data split		TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score
Train	Test								
70	30	17	52	0	1	97,11%	95,94 %	95,94 %	95,94 %
80	20	45	89	3	2	96,40 %	93,75 %	95,74 %	94,73 %
90	10	71	131	3	3	98,57%	100 %	94,44%	97,14 %

In the 70:30 data split scenario, the model achieved an accuracy of 97.11%, with precision and recall of 95.94% each. The F1-score attained a value of 95.94%, signifying a commendable equilibrium between precision and recall. In the 80:20 scenario, the accuracy experienced a marginal decline to 96.40%, accompanied by a precision of 93.75% and a recall of 95.74%. Notwithstanding the marginal reduction in precision, the F1-score persisted at 94.73%, signifying reliable performance. In the interim, the data division in a 90:10 ratio yielded superior accuracy, quantified at 98.57%, with precision attaining a remarkable 100% and recall measuring at 94.44%, culminating in an F1-score of 97.14%.

Overall, the model shows stable performance in various data-splitting scenarios. The 90:10 data partition yields optimal outcomes in terms of accuracy and precision, whereas the 70:30 and 80:20 partitions demonstrate a more favorable equilibrium between precision and recall. These outcomes signify that the model is adept at accurately classifying data, reinforced by the feature selection strategy that employs the Relief algorithm, which aids in recognizing the key features [61]. The model can work more optimally and efficiently in prediction by eliminating less informative features.

According to the information presented in Table 4, the methodology proposed in this research employs the C4.5 algorithm, enhanced by optimization via KNN as the imputation technique and Relief as the method for feature selection. This amalgamation yields a remarkable accuracy rate of 98.57%, suggesting that this strategy can augment the efficacy of pre-existing classification methodologies.

Table 4. Comparison of Classification Accuracy Between Previous Methods and the Proposed Method on WBCD Dataset

No	Authors	Methods	Accuracy
1	[62]	PSO-C4.5	96.49%
2	[63]	EBL-RBFNN	98.40%
3	[64]	Gaussian - Naïve Bayes	97.5%
4	[65]	t-distributed Stochastic Neighbor Embedding (t-SNE)	98.20%
5	Proposed method	Relief-KNN-C4.5	98.57%

Compared to prior investigations, the proposed methodology demonstrates a notable enhancement in accuracy. For instance, the PSO-C4.5 technique utilized by [62] realized an accuracy of 96.49%, whereas the EBL-RBFNN method implemented by [63] attained an accuracy of 98.40%. Moreover, the execution of the Gaussian - Naïve Bayes technique, as specified by [64], in conjunction with the t-distributed Stochastic Neighbor Embedding framework employed by [65], yielded accuracy ratings of 97.5% and 98.20%, correspondingly. Consequently, the methodology proposed in this study is substantiated as superior to earlier techniques, establishing it as a more efficacious option for analyzing the WBCD dataset.

CONCLUSION

This research articulates a refined methodology for the classification of breast cancer by amalgamating the C4.5 algorithm with K-nearest neighbors (KNN) to address the issue of missing value imputation and employing Relief for the selection of pertinent features. The findings underscore that this integration markedly enhances classification accuracy in comparison to antecedent techniques. The advocated method realizes an accuracy enhancement of 2.08% relative to PSO-C4.5, 0.17% in contrast to EBL-RBFNN, 1.07% over Gaussian Naïve Bayes, and 0.37% compared to t-SNE, thereby accentuating its efficacy in refining classification performance for the WBCD dataset.

The findings emphasize the importance of effective feature selection and missing value handling in improving classification accuracy. Utilizing KNN for data imputation, the model proficiently reduces the

loss of information and maintains significant data, whereas the Relief algorithm ascertains that exclusively the most relevant features are engaged in the classification method. This optimization not only improves the accuracy of the model but also diminishes computational intricacies, rendering it a viable and efficient methodology for medical diagnostics.

Given the promising results, future research could explore further enhancements, such as integrating deep learning architectures or hybrid feature selection techniques to improve generalizability across different medical datasets. In addition, the empirical examination of the recommended methodology across more extensive and varied datasets could enhance understanding of its durability and significance in genuine clinical contexts.

REFERENCES

[1] S. Loibl, P. Poortmans, M. Morrow, C. Denkert, and G. Curigliano, "Breast cancer," *The Lancet*, vol. 397, no. 10286, pp. 1750–1769, May 2021, doi: 10.1016/S0140-6736(20)32381-3.

[2] Z. Momenimovahed and H. Salehiniya, "Epidemiological characteristics of and risk factors for breast cancer in the world," *Breast Cancer: Targets and Therapy*, vol. Volume 11, pp. 151–164, Apr. 2019, doi: 10.2147/BCTT.S176070.

[3] A. Sharma, D. Goyal, and R. Mohana, "An ensemble learning-based framework for breast cancer prediction," *Decision Analytics Journal*, vol. 10, 2024, doi: 10.1016/j.dajour.2023.100372.

[4] M. M. Rivera-Franco and E. Leon-Rodriguez, "Delays in Breast Cancer Detection and Treatment in Developing Countries," *Breast Cancer (Auckl)*, vol. 12, p. 117822341775267, Jan. 2018, doi: 10.1177/1178223417752677.

[5] O. Ginsburg *et al.*, "Breast cancer early detection: A phased approach to implementation," *Cancer*, vol. 126, no. S10, pp. 2379–2393, May 2020, doi: 10.1002/cncr.32887.

[6] T. Islam *et al.*, "Predictive modeling for breast cancer classification in the context of Bangladeshi patients by use of machine learning approach with explainable AI," *Sci Rep*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-57740-5.

[7] S. R. S. Chakravarthy, N. Bharanidharan, V. V. Kumar, T. R. Mahesh, M. S. Alqahtani, and S. Guluwadi, "Deep transfer learning with fuzzy ensemble approach for the early detection of breast cancer," *BMC Med Imaging*, vol. 24, no. 1, 2024, doi: 10.1186/s12880-024-01267-8.

[8] M. Rakhshaninejad, M. Fathian, R. Shirkoohi, F. Barzinpour, and A. H. Gandomi, "Refining breast cancer biomarker discovery and drug targeting through an advanced data-driven approach," *BMC Bioinformatics*, vol. 25, no. 1, 2024, doi: 10.1186/s12859-024-05657-1.

[9] J. Guitan, E. L. Snary, M. Arnold, and Y. Chang, "Applications of machine learning in animal and veterinary public health surveillance," *Revue Scientifique et Technique de l'OIE*, vol. 42, pp. 230–241, Jan. 2023, doi: 10.20506/rst.42.3366.

[10] Hemavati, V. S. Devi, and R. Aparna, "Multi-label learning by extended multi-tier stacked ensemble method with label correlated feature subset augmentation," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 3, pp. 3384–3397, 2023, doi: 10.11591/ijece.v13i3.pp3384-3397.

[11] A. Kaltsounis, E. Spiliotis, and V. Assimakopoulos, "Conditional Temporal Aggregation for Time Series Forecasting Using Feature-Based Meta-Learning," *Algorithms*, vol. 16, no. 4, p. 206, Apr. 2023, doi: 10.3390/a16040206.

[12] N. S. F. Putri, A. P. Wibawa, H. A. Rasyid, A. Nafalski, and U. R. Hasyim, "Boosting and bagging classification for computer science journal," *International Journal of Advances in Intelligent Informatics*, vol. 9, no. 1, pp. 27–38, 2023, doi: 10.26555/ijain.v9i1.985.

[13] K. K. Patel, G. Desaulniers, and A. Lodi, "An improved column-generation-based matheuristic for learning classification trees," *Comput Oper Res*, vol. 165, 2024, doi: 10.1016/j.cor.2024.106579.

[14] A. R. Panhalkar and D. D. Doye, "A novel approach to build accurate and diverse decision tree forest," *Evol Intell*, vol. 15, no. 1, pp. 439–453, 2022, doi: 10.1007/s12065-020-00519-0.

[15] L. Meng, B. Bai, W. Zhang, L. Liu, and C. Zhang, "Research on a Decision Tree Classification Algorithm Based on Granular Matrices," *Electronics (Switzerland)*, vol. 12, no. 21, 2023, doi: 10.3390/electronics12214470.

[16] K. Sevvanthi, S. Ganapathy, P. Penumadu, and K. T. Harichandrakumar, "Comparing the predictive performance of a decision tree with logistic regression for oral cavity cancer mortality: A retrospective study," *Cancer Research, Statistics, and Treatment*, vol. 6, no. 1, pp. 103–110, 2023, doi: 10.4103/crst.crst_234_22.

[17] J. Fernandes Andry, H. Tannady, G. Dwinoor Rembulan, and A. Edinata, "Avocado Price Data Analysis Using Decision Tree," *Salud, Ciencia y Tecnología - Serie de Conferencias*, vol. 2, p. 568, Oct. 2023, doi: 10.56294/sctconf2023568.
[18] T. A. Assegie and P. S. Nair, "Handwritten digits recognition with decision tree classification: A machine learning approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4446–4451, 2019, doi: 10.11591/ijece.v9i5.pp4446-4451.
[19] S. Pathan and S. K. Sharma, "Design an Optimal Decision Tree based Algorithm to Improve Model Prediction Performance," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 6, pp. 127–133, 2023, doi: 10.17762/ijritcc.v11i6.7295.
[20] A. A. Alharbi, "Classification Performance Analysis of Decision Tree-Based Algorithms with Noisy Class Variable," *Discrete Dyn Nat Soc*, vol. 2024, pp. 1–10, Feb. 2024, doi: 10.1155/2024/6671395.
[21] U. Ramasamy and S. Sundar, "An Illustration of Rheumatoid Arthritis Disease Using Decision Tree Algorithm," *Informatica (Slovenia)*, vol. 46, no. 1, pp. 109–119, 2022, doi: 10.31449/inf.v46i1.3269.
[22] Imran, M. F. Zuhairi, S. M. Ali, Z. Shahid, M. M. Alam, and M. M. Su'ud, "Realtime Feature Engineering for Anomaly Detection in IoT Based MQTT Networks," *IEEE Access*, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3363889.
[23] Y. Hanyf and H. Silkan, "A method for missing values imputation of machine learning datasets," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 888–898, 2024, doi: 10.11591/ijai.v13.i1.pp888-898.
[24] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *J Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00516-9.
[25] H. Nugroho, N. P. Utama, and K. Surendro, "Normalization and outlier removal in class center-based firefly algorithm for missing value imputation," *J Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00518-7.
[26] A. E. Karrar, "The Effect of Using Data Pre-Processing by Imputations in Handling Missing Values," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 375–384, 2022, doi: 10.52549/ijeei.v10i2.3730.
[27] M.-W. Huang, C.-F. Tsai, S.-C. Tsui, and W.-C. Lin, "Combining data discretization and missing value imputation for incomplete medical datasets," *PLoS One*, vol. 18, no. 11 Novembe, 2023, doi: 10.1371/journal.pone.0295032.
[28] S. Lestari, Y. Yulmaini, A. Aswin, S. Y. Ma'ruf, S. Sulyono, and R. R. N. Fikri, "Alleviating cold start and sparsity problems in the micro, small, and medium enterprises marketplace using clustering and imputation techniques," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, p. 3220, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3220-3229.
[29] S. Mansouri, S. Boulares, and S. Chabchoub, "Machine Learning for Early Diabetes Detection and Diagnosis," *J Wirel Mob Netw Ubiquitous Comput Dependable Appl*, vol. 15, no. 1, pp. 216–230, 2024, doi: 10.58346/JOWUA.2024.I1.015.
[30] T. R. Mahesh, D. Santhakumar, A. Balajee, H. S. Shreenidhi, V. V. Kumar, and J. Rajkumar Annand, "Hybrid Ant Lion Mutated Ant Colony Optimizer Technique With Particle Swarm Optimization for Leukemia Prediction Using Microarray Gene Data," *IEEE Access*, vol. 12, pp. 10910–10919, 2024, doi: 10.1109/ACCESS.2024.3351871.
[31] N. O. Aljehane, H. A. Mengash, S. B. H. Hassine, F. A. Alotaibi, A. S. Salama, and S. Abdelbagi, "Optimizing intrusion detection using intelligent feature selection with machine learning model," *Alexandria Engineering Journal*, vol. 91, pp. 39–49, 2024, doi: 10.1016/j.aej.2024.01.073.
[32] U. Ahmed, A. R. Khan, A. Mahmood, I. Rafiq, R. Ghannam, and A. Zoha, "Short-term global horizontal irradiance forecasting using weather classified categorical boosting," *Appl Soft Comput*, vol. 155, 2024, doi: 10.1016/j.asoc.2024.111441.
[33] Ruchi *et al.*, "Lumbar Spine Disease Detection: Enhanced CNN Model With Improved Classification Accuracy," *IEEE Access*, vol. 11, pp. 141889–141901, 2023, doi: 10.1109/ACCESS.2023.3342064.
[34] A. Jain and V. Jain, "Efficient Framework for Sentiment Classification Using Apriori Based Feature Reduction," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 31, pp. 1–11, 2021, doi: 10.4108/eai.16-2-2021.168715.

[35] W. Zhang and J. Chen, "Relief feature selection and parameter optimization for support vector machine based on mixed kernel function," *International Journal of Performability Engineering*, vol. 14, no. 2, pp. 280–289, 2018, doi: 10.23940/ijpe.18.02.p9.280289.
[36] J. Liu, J. Zhang, Y. Luo, S. Yang, J. Wang, and Q. Fu, "Mass Spectral Substance Detections Using Long Short-Term Memory Networks," *IEEE Access*, vol. 7, pp. 10734–10744, 2019, doi: 10.1109/ACCESS.2019.2891548.
[37] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optim Methods Softw*, vol. 1, no. 1, pp. 23–34, Jan. 1992, doi: 10.1080/10556789208805504.
[38] T. Mladenova and I. Valova, "Classification with K-Nearest Neighbors Algorithm: Comparative Analysis between the Manual and Automatic Methods for K-Selection," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023, doi: 10.14569/IJACSA.2023.0140444.
[39] S. Barrash, Y. Shen, and G. B. Giannakis, "Scalable and Adaptive KNN for Regression Over Graphs," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, IEEE, Dec. 2019, pp. 241–245. doi: 10.1109/CAMSAP45676.2019.9022509.
[40] M. Papanikolaou, G. Evangelidis, and S. Ougiaroglou, "Dynamic k determination in k-NN classifier: A literature review," in *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, Jul. 2021, pp. 1–8. doi: 10.1109/IISA52424.2021.9555525.
[41] S. S. Aung, I. Nagayama, and S. Tamaki, "A High-performance Classifier from K-dimensional Tree-based Dual-kNN," *IEIE Transactions on Smart Processing & Computing*, vol. 7, no. 3, pp. 184–194, Jun. 2018, doi: 10.5573/IEIESPC.2018.7.3.184.
[42] G. Sandhu, A. Singh, P. S. Lamba, D. Virmani, and G. Chaudhary, "Modified Euclidean-Canberra blend distance metric for kNN classifier," *Intelligent Decision Technologies*, vol. 17, no. 2, pp. 527–541, May 2023, doi: 10.3233/IDT-220233.
[43] G. Sun, Y. Sun, and F. Luo, "Generalized Minkowski Distance-Based Local Mean k-Nearest Neighbor Classifier," in *2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA)*, IEEE, Aug. 2023, pp. 1710–1716. doi: 10.1109/ICIPCA59209.2023.10257763.
[44] F. Moslehi and A. Haeri, "An evolutionary computation-based approach for feature selection," *J Ambient Intell Humaniz Comput*, vol. 11, no. 9, pp. 3757–3769, Sep. 2020, doi: 10.1007/s12652-019-01570-1.
[45] N. Bacanin, K. Venkatachalam, T. Bezdan, M. Zivkovic, and M. Abouhawwash, "A novel firefly algorithm approach for efficient feature selection with COVID-19 dataset," *Microprocess Microsyst*, vol. 98, p. 104778, Apr. 2023, doi: 10.1016/j.micpro.2023.104778.
[46] M. Moran and G. Gordon, "Deep Curious Feature Selection: A Recurrent, Intrinsic-Reward Reinforcement Learning Approach to Feature Selection," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 3, pp. 1174–1184, Mar. 2024, doi: 10.1109/TAI.2023.3282564.
[47] A. Benkessirat and N. Benblidia, "Fundamentals of Feature Selection: An Overview and Comparison," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, IEEE, Nov. 2019, pp. 1–6. doi: 10.1109/AICCSA47632.2019.9035281.
[48] Y. Kaushik, M. Dixit, N. Sharma, and M. Garg, "Feature Selection Using Ensemble Techniques," 2021, pp. 288–298. doi: 10.1007/978-981-16-1480-4_25.
[49] D. Hemavathi and H. Srimathi, "An efficient approach to identify an optimal feature selection method using improved principle component analysis in supervised learning process," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 7 Special issue, 2019.
[50] E. C. Blessie and E. Karthikeyan, "RELIEF-DISC: An Extended RELIEF Algorithm Using Discretization Approach for Continuous Features," in *2011 Second International Conference on Emerging Applications of Information Technology*, IEEE, Feb. 2011, pp. 161–164. doi: 10.1109/EAIT.2011.39.
[51] D. R. Munirathinam and M. Ranganadhan, "A new improved filter-based feature selection model for high-dimensional data," *J Supercomput*, vol. 76, no. 8, pp. 5745–5762, Aug. 2020, doi: 10.1007/s11227-019-02975-7.
[52] N. Aggarwal *et al.*, "Mean based relief: An improved feature selection method based on ReliefF," *Applied Intelligence*, vol. 53, no. 19, pp. 23004–23028, Oct. 2023, doi: 10.1007/s10489-023-04662-w.

[53] G. Varshini, A. Ramya, C. L. Sravya, V. Kumar, and B. K. Shukla, "Improving Heart Disease Prediction of Classifiers with Data Transformation using PCA and Relief Feature Selection," in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, IEEE, Mar. 2023, pp. 1644–1649. doi: 10.1109/ICEARS56392.2023.10085401.

[54] Y. Diao and Q. Zhang, "Optimization of Management Mode of Small- and Medium-Sized Enterprises Based on Decision Tree Model," *Journal of Mathematics*, vol. 2021, pp. 1–9, Dec. 2021, doi: 10.1155/2021/2815086.

[55] J. Xu, "Refining and implementing of a decision tree based risk assessment model for college students' innovation and entrepreneurship," *Journal of Computational Methods in Sciences and Engineering*, vol. 24, no. 4–5, pp. 3093–3111, Aug. 2024, doi: 10.3233/JCM-247556.

[56] H. Mardiansyah, M. Zarlis, and O. S. Sitompul, "Analysis of C4.5 Algorithm of Water Quality Dataset," *J Phys Conf Ser*, vol. 1898, no. 1, p. 012002, Jun. 2021, doi: 10.1088/1742-6596/1898/1/012002.

[57] O. Loyola-González, E. Ramírez-Sáyago, and M. A. Medina-Pérez, "Towards improving decision tree induction by combining split evaluation measures," *Knowl Based Syst*, vol. 277, p. 110832, Oct. 2023, doi: 10.1016/j.knosys.2023.110832.

[58] A. Argente-Garrido, C. Zuheros, M. V. Luzón, and F. Herrera, "An interpretable client decision tree aggregation process for federated learning," *Inf Sci (N Y)*, vol. 694, p. 121711, Mar. 2025, doi: 10.1016/j.ins.2024.121711.

[59] I. Barranco-Chamorro and R. M. Carrillo-García, "Techniques to deal with off-diagonal elements in confusion matrices," *Mathematics*, vol. 9, no. 24, 2021, doi: 10.3390/math9243233.

[60] C. S. Hong and T. G. Oh, "TPR-TNR plot for confusion matrix," *Commun Stat Appl Methods*, vol. 28, no. 2, 2021, doi: 10.29220/CSAM.2021.28.2.161.

[61] Najmusseher and M. Umme Salma, "Impact of Feature Selection Techniques for EEG-Based Seizure Classification," 2023, pp. 197–207. doi: 10.1007/978-981-19-9379-4_16.

[62] M. A. Muslim, A. Nurzahputra, and B. Prasetiyo, "Improving accuracy of C4.5 algorithm using split feature reduction model and bagging ensemble for credit card risk prediction," in *2018 International Conference on Information and Communications Technology, ICOIACT 2018*, 2018. doi: 10.1109/ICOIACT.2018.8350753.

[63] A. H. Osman and H. M. A. Aljahdali, "An Effective of Ensemble Boosting Learning Method for Breast Cancer Virtual Screening Using Neural Network Model," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2976149.

[64] M. V. Anand, B. Kiranbala, S. R. Srividhya, K. C., M. Younus, and M. H. Rahman, "Gaussian Naïve Bayes Algorithm: A Reliable Technique Involved in the Assortment of the Segregation in Cancer," *Mobile Information Systems*, vol. 2022, 2022, doi: 10.1155/2022/2436946.

[65] N. Sharma, K. P. Sharma, M. Mangla, and R. Rani, "Breast cancer classification using snapshot ensemble deep learning model and t-distributed stochastic neighbor embedding," *Multimed Tools Appl*, vol. 82, no. 3, 2023, doi: 10.1007/s11042-022-13419-5.