Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 1, January 2026, pp: 368-394 e-ISSN: 2656-8632

,RESEARCH ARTICLE

OPEN ACCESS

A Comparative Analysis of SMOTE and ADASYN for Cervical Cancer Detection using XGBoost with MICE Imputation

Mita Azzahra Ramadhan, Triando Hamonangan Saragih, Dwi Kartini , Muhammad Itqan Mazdadi, and Muliadi

Computer Science Department, Lambung Mangkurat University, Banjarbaru, South Kalimantan, Indonesia

Corresponding author: Triando Hamonangan Saragih (e-mail: triando.saragih@ulm.ac.id)

Abstract Cervical cancer remains a significant global health burden for women, with approximately 660,000 new cases and 350,000 associated deaths recorded worldwide in 2022. Machine learning methods have shown great promise in advancing timely detection and accurate diagnosis. This investigation compares two widely used oversampling strategies, Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN), applied to cervical cancer identification via the XGBoost classifier, paired with Multiple Imputation by Chained Equations (MICE) to handle incomplete data. The dataset consists of cervical cancer risk factors with four diagnostic outcomes: Hinselmann, Schiller, Cytology, and Biopsy, which are treated as independent binary classification tasks rather than a single multilabel classification problem. The process began by preparing a dataset of cervical cancer risk factors through MICE imputation, then applying SMOTE and ADASYN to address class imbalance. The XGBoost model is optimized using Random Search hyperparameter tuning and evaluated across train-test split ratios (50:50, 60:40, 70:30, 80:20, and 90:10) using accuracy, precision (macro, micro, weighted), recall (macro, micro, weighted), F1-score (macro, micro, weighted), and AUC metrics. The results indicated that the XGBoost setup with MICE and SMOTE outperformed the others, achieving 97.1% accuracy, 97.1% mic-precision, 97.1% mic-recall, 97.1% mic-F1, and 97.1% AUC. Meanwhile, the ADASYN-integrated model showed marginally lower results, with 95.4% accuracy, 95.4% micro-precision, 95.4% micro-recall, 95.4% micro-F1, and 55.5% AUC. SMOTE proved more adept at creating evenly distributed synthetic data for the underrepresented group. Overall, this work underscores the value of integrating MICE imputation, SMOTE oversampling, and tuned XGBoost as a reliable approach for cervical cancer detection. These insights pave the way for automated screening tools that can bolster clinical judgment and improve early diagnosis outcomes.

Keywords Cervical cancer; MICE; XGBoost; Random Search; SMOTE; ADASYN.

I. Introduction

The cervix is part of the female reproductive organ and is located at the lower fibromuscular portion of the uterus [1]. When the cells that cover the cervix start to grow and multiply uncontrollably without following the proper mechanisms, it can lead to cervical cancer [2]. Cervical cancer is one of the leading causes of women's deaths worldwide [3]. In 2022, there will be an estimated 660,000 new cases and 350,000 deaths due to cervical cancer worldwide [4]. Cervical cancer is caused by the human papillomavirus (HPV), with the highest risk types being HPV 16, 18, 31, and 33 [5]. HPV is mostly transmitted through sexual contact and targets basal keratinocytes in the genital mucosa, oral mucosa, and skin [6]. Various factors, such as smoking, long-term use of oral contraceptives, multiple pregnancies, or pregnancy at a young age, may also increase the risk of cervical cancer [7]. Treatments for cervical cancer include radiation, surgery, chemotherapy [8], and Pap smears. Pap smears have been the lifesaver for millions of women with cervical cancer [9].

Cervical cancer can be detected with machine learning. ML models have been shown to accelerate the diagnosis of cervical cancer [2]. Research on cervical cancer has been conducted by [2] using the decision tree method, with features selected via RFE and SMOTE-Tomek, achieving an accuracy of 98.72% and a sensitivity of 100%. Other research on cervical cancer was conducted by [10] using the Random Forest, Decision Tree, Adaptive Boosting, and Gradient Boosting methods with an accuracy value of 100%, while the SVM method produced an accuracy value of 99%. Although these results are encouraging, selecting an appropriate model remains crucial, particularly for medical datasets that are often characterized by class imbalance, complex feature interactions, and a high risk of overfitting. Consequently, ensemble learning

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415
Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

methods that employ regularization and advanced optimization algorithms are increasingly favored in cancer detection tasks. One of the machine learning models commonly used in the medical field, including detecting cancer, is called XGBoost [11]. XGBoost is an extension of GBDT [12] that is efficient for complex classification tasks [13]. Moreover, XGBoost employs advanced hyperparameter tuning mechanisms to reduce overfitting, decrease prediction variance, and improve model accuracy, thereby enabling optimal model performance [14][15]. Several studies have demonstrated the effectiveness of XGBoost in medical applications. For instance, research by [16] using the XGBoost method on heart disease achieved an accuracy value of 91.8%. Other research conducted by [17] on Aneurysmal Subarachnoid Hemorrhage disease resulted in higher auc values in the XGBoost model to predict mortality and adverse functional outcomes, which were 0.950 and 0.958, compared to logistic regression models, which were 0.767 and 0.829.

To improve the performance of the XGBoost model, hyperparameter tuning using Random Search is employed to identify configurations that deliver optimal outcomes. Hyperparameter optimization has been shown to significantly influence the performance of machine learning models, although its effectiveness may vary across algorithms [18]. Compared to exhaustive methods such as Grid Search, RS samples hyperparameters randomly from predefined distributions, enabling broader, more diverse exploration of the search space at substantially lower computational cost [19]. Although more advanced approaches, such as Bayesian Optimization, provide adaptive search strategies, they often involve higher computational complexity and require careful modeling of the objective function, which may be less practical for high-dimensional hyperparameter spaces. Consequently, RS represents a suitable trade-off between computational efficiency and search diversity, particularly for complex models such as XGBoost. Despite its advantages, RS is not without limitations, as its stochastic sampling strategy may lead to variability in outcomes [20]. Nevertheless, several studies have demonstrated the effectiveness of RS in improving the performance of XGBoost models. For instance, [21] applied XGBoost with RS to the Chronic Kidney Failure dataset from the UCI Machine Learning Repository and combined XGBoost with Random Search, achieving an accuracy of 98.57% and an F-Measure of 0.9842. In another study, [22] applied XGBoost optimized with RS for shallow landslide classification in Trabzon Province, Turkey, reporting an accuracy of 75.64%, precision of 94.71%, recall of 55.23%, and an F1-score of 69.77%.

One of the common challenges in machine learning based disease detection is the presence of missing values and class imbalance. The cervical cancer dataset used in this study, obtained from the UCI Machine Learning Repository, contains a substantial proportion of missing data, which can adversely affect model accuracy and predictive reliability if not properly addressed [23]. Although missing values can be handled using various imputation techniques [24] simple approaches such as mean, median, or mode imputation may distort data distributions and fail to preserve important relationships among features, which are critical in medical datasets. Therefore, this study employs Multivariate Imputation by Chained Equations (MICE), a model-based imputation method that iteratively imputes missing values by modeling each feature conditional on the others [25] [26]. MICE is particularly advantageous for the cervical cancer dataset, as it is capable of maintaining complex inter-feature relationships and reducing bias introduced by missing data, thereby producing more realistic and statistically consistent imputations. The effectiveness of MICE has been demonstrated in previous studies, for example, [27] using MICE techniques to replace outliers in concrete slump data from the UCI Machine Learning Repository resulted in the highest value in the R^2 stacking model 0.9702, RMSE and MAE of the KNN model are 0.1392 and 0.1162.

Imbalanced data makes it difficult for machine learning models to perform optimally [28]. Oversampling is a commonly used strategy to address this issue by increasing the number of minority class samples to achieve a more balanced class distribution [29] [30]. Among various oversampling approaches, Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN) were selected in this study due to their distinct mechanisms for generating synthetic minority samples. SMOTE creates synthetic instances through interpolation between existing minority samples and their nearest neighbors, aiming to achieve a more uniform class balance [31]. In contrast, ADASYN adaptively generates synthetic samples by emphasizing minority instances that are more difficult to learn, particularly those located in regions with high class overlap [32]. These fundamental differences provide a strong basis for comparative analysis, as SMOTE focuses on global class balance, whereas ADASYN prioritizes learning from challenging samples. Research conducted by [33] using the SMOTE and ADASYN methods to balance data on CCF datasets resulted in the highest accuracy value in the Random Forest model with SMOTE, which is 99.99%, while the accuracy value of Random Forest with ADASYN is 99.98%.

The primary objective of this research is to assess and compare the efficacy of oversampling methods for classifying cervical cancer using an XGBoost model. Addressing challenges of missing data and class imbalance, the study explores the relative effects of SMOTE and ADASYN, combined with MICE

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

imputation, on classification outcomes, with the aim of determining the optimal strategy to enhance predictive accuracy in unbalanced cervical cancer datasets. This work provides several contributions: it develops an integrated evaluation framework that merges MICE imputation, oversampling approaches, and XGBoost modeling; it conducts a thorough comparison of SMOTE and ADASYN within a unified classification workflow, offering insights into their comparative efficacy in managing class imbalance; and it analyzes the influence of these preprocessing methods on classification results, emphasizing the importance of data preprocessing and model refinement in boosting predictive reliability. The key contributions of this study are outlined as follows:

1) An integrated classification framework designed for cervical cancer datasets, incorporating MICE imputation, oversampling methods, and XGBoost enhanced by Random Search for hyperparameter optimization.
2) A comprehensive comparative study of SMOTE and ADASYN oversampling techniques, emphasizing their effects on classification outcomes within a consistent modeling setup.
3) An empirical assessment identifying the most efficient oversampling approach to enhance cervical cancer classification in unbalanced datasets.

The subsequent sections of this paper are organized as follows. Section I surveys existing literature on cervical cancer classification and techniques for addressing data imbalance. Section II details the dataset, preprocessing procedures, oversampling methods, and the proposed XGBoost-based approach. Section III reports the experimental outcomes and performance metrics. Section IV deliberates on the results and the study's constraints. Lastly, Section V summarizes the conclusions and suggests avenues for future investigation.

II. Method

This research compares SMOTE and ADASYN on cervical cancer classification using the XGBoost method. The flowchart of this research is represented in Fig.. This research uses Python for machine learning classification on the “Cervical Cancer” dataset from UCI. The methodology consists of several stages. First, data preprocessing using the MICE (Multivariate Imputation by Chained Equations) technique was performed to impute missing values. The dataset is then divided into training and testing sets in various proportions (90:10, 80:20, 70:30, 60:40, and 50:50) for comparative analysis. Next, the data was balanced using SMOTE and ADASYN. The data is then classified using the XGBoost method with Randomized Hyperparameter Tuning. Finally, the test data evaluation results are measured using accuracy, precision (macro, micro, and weighted), recall (macro, micro, and weighted), F1-score (macro, micro, and weighted), and AUC.

A. Data Collection

The dataset used in this study is the Cervical Cancer (Risk Factors) Dataset from the University of California, Irvine (UCI) Machine Learning Repository, which contains patient data from “Hospital Universitario de Caracas” in Caracas, Venezuela. This dataset can be accessed through the following link: https://archive.ics.uci.edu/dataset/383/cervical+cancer

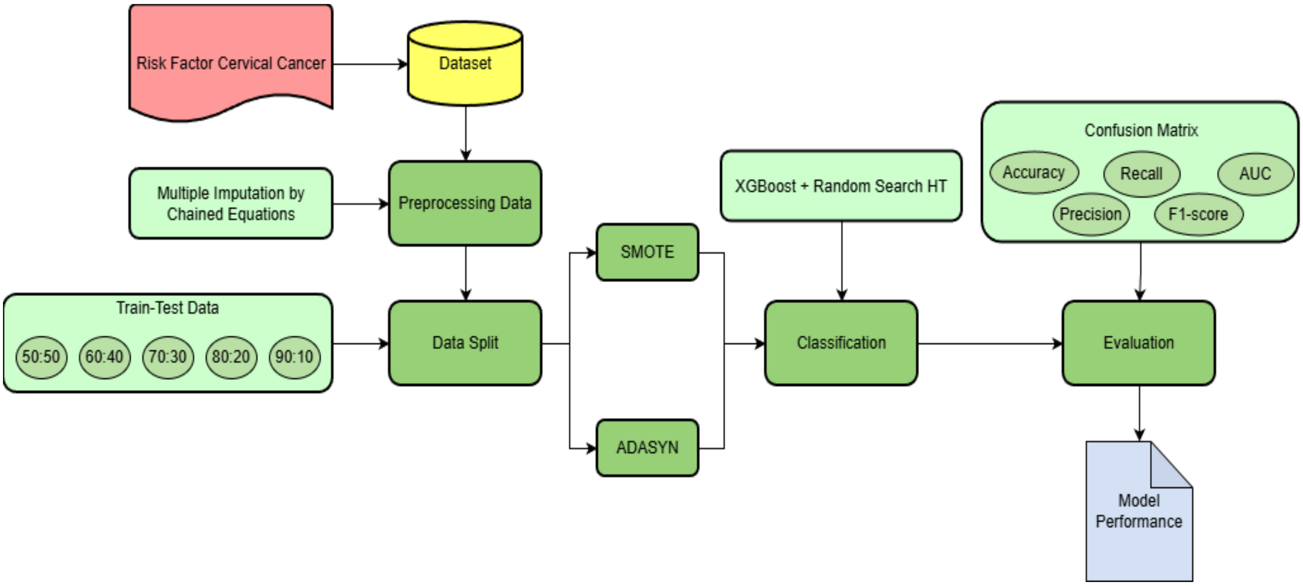

Fig. 1. **Research flowchart comparing SMOTE and ADASYN in cervical cancer classification, with MICE imputation and XGBoost optimization using Random Search hyperparameter tuning.**

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Table 1. A detailed description of the Cervical Cancer dataset used in the current study

No	Attribute	Type	Range	Missing Value
1	Age	Integer	13 - 84 Years	0
2	Number of sexual partners	Integer	1 - 28 sexual partners	26
3	First sexual intercourse	Integer	10 - 32 years	7
4	Number of pregnancies	Integer	0 - 11 pregnancies	56
5	Smokes	Boolean	0 → No, 1 → Yes	13
6	Smokes (years)	Integer	0 - 37 years	13
7	Smokes (pack/year)	Integer	0 - 37 packs per years	13
8	Hormonal Contraceptives	Boolean	0 → No, 1 → Yes	108
9	Hormonal Contraceptives (years)	Integer	0 - 30 years	108
10	Intrauterine Device (IUD)	Boolean	0 → No, 1 → Yes	117
11	IUD (years)	Integer	0 - 19 years	117
12	Sexually Transmitted Diseases (STDs)	Boolean	0 → No, 1 → Yes	105
13	STDs (number)	Boolean	0 → No, 1 → Yes	105
14	STDs:condylomatosis	Boolean	0 → No, 1 → Yes	105
15	STDs:cervical condylomatosis	Boolean	0 → No, 1 → Yes	105
16	STDs: vaginal condylomatosis	Boolean	0 → No, 1 → Yes	105
17	STDs: vulvo-perineal condylomatosis	Boolean	0 → No, 1 → Yes	105
18	STDs:syphilis	Boolean	0 → No, 1 → Yes	105
19	STDs:pelvic inflammatory disease	Boolean	0 → No, 1 → Yes	105
20	STDs:genital herpes	Boolean	0 → No, 1 → Yes	105
21	STDs:molluscum contagiosum	Boolean	0 → No, 1 → Yes	105
22	STDs:AIDS	Boolean	0 → No, 1 → Yes	105
23	STDs:HIV	Boolean	0 → No, 1 → Yes	105
24	STDs:Hepatitis B	Boolean	0 → No, 1 → Yes	105
25	STDs:HPV	Boolean	0 → No, 1 → Yes	105
26	STDs: Number of diagnoses	Integer	0 - 3	0
27	STDs: Time since first diagnosis	Integer	1 - 22	787
28	STDs: Time since last diagnosis	Integer	1 - 22	787
29	Dx:Cancer	Boolean	0 → No, 1 → Yes	0
30	Dx:CIN	Boolean	0 → No, 1 → Yes	0
31	Dx:HPV	Boolean	0 → No, 1 → Yes	0
32	Diagnosis: Dx	Boolean	0 → No, 1 → Yes	0
33	Hinselmann (Target Variable)	Boolean	0 → No, 1 → Yes	0
34	Schiller (Target Variable)	Boolean	0 → No, 1 → Yes	0
35	Citology (Target Variable)	Boolean	0 → No, 1 → Yes	0
36	Biopsy (Target Variable)	Boolean	0 → No, 1 → Yes	0

+risk+factors. This dataset was chosen due to its comprehensive representation of cervical cancer risk factors, which contains 858 patient data and 36 attributes. The attributes include patient demographic information, medical history, and lifestyle factors. The target variables are the diagnosis results of Hinselmann, Schiller, Cytology, and Biopsy, which are the main diagnosis methods for cervical cancer. Details of the dataset's features are presented in Table.

B. Multiple Imputation by Chained Equations (MICE)

MICE is an imputation method that fills in missing data using a univariate conditional distribution for each

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

variable, considering the other variables iteratively [34]. MICE, also known as Multiple Sequential Regression Imputation, was first introduced by Donald Bruce Rubin in 1987 [35]. The development and refinement of this method was then popularized more widely by Stef Van Buuren in the early 2000s through his contributions in the field of statistics and the development of software supporting this technique [27].

The advantages of MICE lie in its ability to consider uncertainty in imputation [36], use relationships between variables [25], and provide flexibility for imputing continuous, binary, and categorical data with regression models appropriate for each type of data [37] [38] [26]. The steps of the MICE algorithm are as follows:

1. Initial Imputation
   Replace missing values using simple methods, such as mean imputation, to generate an initial complete data set [39].
2. Storage in MIDS Object
   Store the initial imputed dataset in an object called Multiply Imputed Dataset (MIDS), which is a copy of the original dataset with the missing values replaced [40].
3. Regression and Coefficient Estimation
   Perform an Ordinary Least Squares (OLS) regression on each of the imputed datasets. From this regression, regression coefficients are obtained, which are used to estimate the missing values. The regression results are stored in the Multiply Imputed Repeated Analysis (MIRA) object shown in Eq. (1) as follows [41]:

   $$X_1 = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon \qquad (1)$$

   where, X_1 is the dependent variable that the model wants to predict or explain. The term β_0 represents the baseline value of X_1 when all independent variables $X_2, X_3, \ldots, X_k$ are zero. The coefficients $\beta_1, \beta_2, \ldots, \beta_k$ indicate the effect of each independent variable, which serves as a predictor, on the dependent variable. The term ε accounts for model error by capturing factors not explained by the predictors.

   Suppose each of the k independent variables, $x_1, x_2, \ldots, x_k$, has n levels. Then, x_{ij} is the i^{th} level of the j^{th} independent variable x_j and $y_1, y_2, \ldots, y_n$ has n levels. Thus, the n-tuples of observations are assumed to follow the same model, which is expressed in Eq. (2) to Eq. (5) below:

   $$y_1 = b_0 + b_1 x_{11} + b_2 x_{12} + \cdots + b_k x_{1k} + e_1 \qquad (2)$$
   $$y_2 = b_0 + b_1 x_{21} + b_2 x_{22} + \cdots + b_k x_{2k} + e_2 \qquad (3)$$
   $$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_k x_{ik} + e_i \qquad (4)$$
   $$\cdots\cdots\cdots$$
   $$y_n = b_0 + b_1 x_{n1} + b_2 x_{n2} + \cdots + b_k x_{nk} + e_n \qquad (5)$$

   If each independent variable has n observations, this Eq. (6) can be written in matrix form as follows [42]:

   $$y = X\beta + \varepsilon \qquad (6)$$

   where X is an $(n \times k)$ matrix of n observations on k independent variables $X_1, X_2, \ldots, X_k, , y$ is an $(n \times 1)$ vector of n observations of the research variable, β is a $(k \times 1)$ vector of regression coefficients and ε is an $(n \times 1)$ vector of disturbances. Using matrix notation, this equation can be written Eq. (7) below:

   $$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1n} \\ 1 & x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nn} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_n \end{bmatrix} \qquad (7)$$

   The regression coefficient can be calculated using the formula in Eq. (8) [42]:

   $$\beta = (X'X)^{-1}X'y \qquad (8)$$

   where X' is the transpose matrix of X.
4. Pooling Estimates
   Pool all coefficient estimates from the imputed dataset using Rubin's Rules. The pooled mean estimate is calculated in Eq. (9) [43]:

   $$\hat{\theta}_{combined} = \frac{1}{m}\sum_{i=1}^{m}\hat{\theta}_i \qquad (9)$$

   where, $\hat{\theta}_{combined}$ is calculated by averaging the mean estimates $\hat{\theta}_i$ from each of the m is the number of imputed datasets, providing a single representative value that accounts for variability introduced by the imputation process.

   After that, perform a combined variance that takes into account the variance of each estimate and the variance between imputations in Eq. (10) [44]:

   $$V_{\theta_{combined}} = \frac{1}{m}\sum_{i=1}^{m} V\left(\hat{\theta}_i\right) + \left(1 + \frac{1}{m}\right).B(\theta) \qquad (10)$$

   where the combined variance, denoted as $V_{\theta_{combined}}$, is obtained from multiple imputation by accounting for both within- and between-imputation variability. In this context, m represents the number of imputations used (for example, if there are 5 imputation datasets, then $m = 5$), while $\hat{\theta}_i$ denotes the parameter estimate of the i^{th} imputation dataset, such as a mean or regression coefficient. The term $V\left(\hat{\theta}_i\right)$ indicates within-dataset variance, reflecting the uncertainty in the estimate for the dataset, and $B(\theta)$ represents the between-imputation variance, capturing the variation between estimates from different imputation datasets.
5. Process Iteration
   Repeat steps 2 to 4 for each variable with missing data. One-time processing of all variables is referred to as one cycle (iteration). This cycle is repeated several times until the imputation results are stable, meaning the predicted values change little between cycles.

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Table 2. A representative sample of the cervical cancer dataset before data preprocessing steps

Age	Number of sexual partners	First sexual intercourse	...	Hinselmann	Schiller	Citology	Biopsy
18	4	15	...	0	0	0	0
15	1	14	...	0	0	0	0
34	1	?	...	0	0	0	0
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
32	2	19	...	0	0	0	0
25	2	17	...	0	0	1	0
33	2	24	...	0	0	0	0
29	2	20	...	0	0	0	0

Table 3. A representative sample of the cervical cancer dataset after data preprocessing steps.

Age	Number of sexual partners	First sexual intercourse	...	Hinselmann	Schiller	Citology	Biopsy
18	4	15	...	0	0	0	0
15	1	14	...	0	0	0	0
34	1	202.147	...	0	0	0	0
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
32	2	19	...	0	0	0	0
25	2	17	...	0	0	1	0
33	2	24	...	0	0	0	0
29	2	20		0	0	0	0

Table presents the “Cervical Cancer” sample dataset before preprocessing, and Table presents the “Cervical Cancer” sample dataset after preprocessing.

C. Oversampling

Unbalanced datasets lead to imbalances in the minority and majority classes in multiclass classification problems. Unbalanced data occurs when the minority class has fewer samples than the majority class. Class imbalance can adversely affect model training, degrade classification performance, and lead to high false positives for certain minority class samples [45] [46]. Oversampling is a technique for overcoming imbalanced datasets [29] that is easily applied to multiclass classification [47] which works by replicating samples from the minority class [48], thus increasing the size of the dataset [33] making the number of samples equal to the majority class [49]. Oversampling does not require extensive parameter tuning and can be performed in seconds [31]. Oversampling can balance the class distribution of a dataset without losing information [50], although it can lead to overfitting [32] due to the large number of replicated samples in the minority class and cannot contribute to extending the decision boundary to the majority class region [28]. Therefore, oversampling has several variations to overcome overfitting. SMOTE and ADASYN are oversampling techniques often used to balance datasets and avoid overfitting.

1. Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is an oversampling technique that starts by randomly selecting a minority class instance and finding its k-nearest minority class neighbors [51].

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v8i1.1415
Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

SMOTE is an improved method based on Random Oversampling (ROS) [52] that was first proposed by Narasimhan Chawla and colleagues in 2002 [53]. SMOTE overcomes overfitting by generating synthetic samples that are similar, but not identical, to the minority class samples [50]. This technique improves the representation of rare events, allowing the model to learn patterns in imbalanced data [54], and improves the accuracy of minority class fault detection [46]. The algorithm of SMOTE is as follows:

a) Calculating Euclidean Distance and Finding K Nearest Neighbors

For each sample in the training set, calculate the Euclidean distance to each minority class sample x_i, and find the k nearest neighbors for each minority-class sample. The Euclidean distance between two points in the feature space $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ is calculated by the following Eq. (11) [55]:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{11}$$

where the distance $d(x, y)$ represents the measure of dissimilarity between two data points x and y. Each x_i dan y_i corresponds to the i^{th} feature of the sample x and y, respectively, and n denotes the total number of features considered in the calculation.

b) Determine the Sampling Ratio (N)

Based on the sampling imbalance rate, determine the sampling ratio N. For x_i, randomly select N samples from its k nearest neighbors, denoted as x_h [30].

c) Building a New Synthetic Sample

Build a new sample based on x_i and x_h until the classes are balanced, denoted as X_{new} in the following Eq. (12) [56]:

$$X_{new} = x_i + \lambda . (x_h - x_i) \tag{12}$$

where a new synthetic sample, denoted as X_{new}, is generated based on an original minority sample, x_i, and one of its nearest neighbors, x_h. The interpolation between x_i and x_h is controlled by a random value λ ranging from 0 to 1, which determines the relative contribution of the original sample and its neighbor in creating the synthetic instance.

2. Adaptive Synthetic Sampling (ADASYN)

ADASYN is an extension of the SMOTE method that serves as an adaptive oversampling technique for minority classes [57]. By generating new synthetic samples around hard-to-classify data, ADASYN increases data variability, reduces the risk of overfitting, and is proven to improve machine learning model performance on highly imbalanced datasets [58]. ADASYN uses a density distribution to adaptively generate a number of synthetic samples, whereas SMOTE generates the same number of synthetic samples for each minority class [59]. The steps of applying the ADASYN technique are as follows:

1. Class Initialization and Ratio

The first step in ADASYN is to calculate the ratio between the number of samples of the minority class (N_{min}) and the majority class $\left(N_{maj}\right)$ in Eq. (13):

$$T = \frac{N_{min}}{N_{maj}} \tag{13}$$

The value of T is used for algorithm initialization and as a basis for calculating the oversampling requirement [60].

2. Determining the Total Number of Synthetic Samples (G)

Based on the degree of imbalance and the parameter β (usually between 0 - 1), calculate the total number of synthetic samples to be generated in Eq. (14).

$$G = \left(\left|N_{maj}\right| - |N_{min}|\right) \times \beta \tag{14}$$

where $\beta \epsilon [0, 1]$ is a parameter that represents the desired level of balance after adding synthetic data. A value of $\beta = 0$ means that no synthetic samples are added, while $\beta = 1$ will result in a fully balanced dataset, where the majority and minority classes have equal proportions [61].

3. Calculate Majority Dominance Around Each Minority Sample

For each minority sample x_i, the ADASYN algorithm finds the k nearest neighbors based on Euclidean distance, then calculates the majority dominance ratio r_i in Eq. (15) as follows [62]:

$$r_i = \frac{\Delta_i}{k} \tag{15}$$

Where the majority dominance rate r_i, represents the proportion of the majority class sample surrounding the i^{th} minority sample. In this context, Δ_i denotes the number of majority neighbors of that minority sample, while k indicates the total number of neighbors considered when calculating the dominance rate.

4. Normalization of r_i

The r_i values are then normalized to produce a weight distribution G_i that sums to 1 in Eq. (16) [63]:

$$G_i = \frac{r_i}{\sum_{i=1}^{N_{min}} r_i} \tag{16}$$

where the normalized weight for the i^{th} minority sample, G_i, is determined based on the majority dominance ratio r_i of that sample. N_{min} represents the total number of minority samples, and the sum of all majority dominance ratios across these samples, $\sum_{i=1}^{N_{min}} r_i$, reflects the overall influence of the majority class in the neighborhood of the minority samples.

5. Determining the Number of Synthetic Samples per Point (g_i)

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

The number of synthetic samples that need to be made from each minority sample x_i in Eq. (17) as follows [64]:

$$g_i = G_i \times G \tag{17}$$

where the number of synthetic samples to be created around the i^{th} minority sample is denoted as g_i, and this is determined based on the normalized weight G_i of that sample. The total number of synthetic samples created across all minority samples is represented by G, providing a measure of the overall augmentation applied to the dataset.

6. Creating Synthetic Samples

For each minority sample x_i, g_i a synthetic sample is created by interpolating one of its neighbors x_{zi} in Eq. (18) as follows [65]:

$$s_i = x_i + \lambda \cdot (x_{zi} - x_i) \tag{18}$$

where a synthetic data point, denoted as s_i, is generated based on an original processed minority sample x_i and one of its nearest neighbors x_{zi} from the minority class. The interpolation between x_i and x_{zi} is controlled by a random value λ ranging from 0 to 1, which determines the relative contribution of the original sample and its neighbor in creating the synthetic instance.

D. Data Split

Before classification, the dataset is split into training and test sets. The machine learning model is trained on training data and evaluated on test data. In this study, the training and testing data are divided into several proportions: 90:10, 80:20, 70:30, 60:40, and 50:50 [66], [67].

E. Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized and highly scalable decision tree-based machine learning algorithm [68]. It was developed by Chen and Guestrin as a highly scalable end-to-end boosting system that has been widely implemented and optimized in various research fields [69]. XGBoost is an extension of the Gradient Boosted Regression Trees (GBRT) framework designed to deliver high prediction performance. As an ensemble learning method, XGBoost combines prediction results from a number of weak models to form a stronger model.

One of the key features of XGBoost is the use of objective function normalization to reduce model complexity, speed up the training process, and reduce the risk of overfitting. Empirically, XGBoost performs relatively faster than other ensemble classification algorithms. It also supports parallel processing so that it can utilize multicore computer resources to efficiently handle large datasets [70]. The following are the steps of the XGBoost algorithm:

1. Boosting Model Prediction

The XGBoost model consists of K CART trees, with the output being the sum of their outputs. The cumulative value serves as the predictive value of the XGBoost model and can be expressed mathematically in Eq. (19) as follows [12]:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{19}$$

where the number of CART trees usd in the model is denoted by K, and f_k refers to a specific CART tree within the ensemble. The output of the XGBoost model for a given input is represented by $\hat{y}_i$, which aggregates the contributions of all individual trees in the ensemble.

2. Loss and Regularization Functions

XGBoost is similar to most machine learning models, and its objective function can be the sum of a loss function and a regular term, which control the accuracy and complexity of the model, respectively. The specific Eq. (20) is as follows [71]:

$$L(\hat{y}) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega\,(f_k) \tag{20}$$

Where:

$$\Omega\left(f_k\right) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{21}$$

where the target value for a given observation is denoted as y_i, and the loss function, l, measure the difference between the predicted value $\hat{y}_i$ and y_i. The regularization term, Ω, is included to penalize model complexity and avoid overfitting. In the decision tree, T represents the number of leaves, with w_j denoting the weight of the j^{th} leaf. The parameter γ penalizes the number of leaves, while λ serves as a regularization parameter for leaf weights, ensuring the model maintains both accuracy and generalization.

3. Taylor Expansion for Model Updating

GB is effective in regression and classification problems. GB is used with a loss function, which is expanded by a second-order Taylor expansion, with constant terms removed to produce a simplified objective in the first step, in Eq. (22) to Eq. (24) as follows [72]:

$$L^{(t)} = \sum_{i=1}^{n}\left(g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)\right) + \Omega(f_k) \tag{22}$$

$$= \sum_{i=1}^{n}\left(g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)\right) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{23}$$

$$= \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right) w_j^2\right] + \gamma T \tag{24}$$

where $I_j = \{i | q(x_i) = j\}$ denotes the sample set of leaf t, and

$$g_i = \frac{\partial l\left(\hat{y}_i^{(t-1)}, y_i\right)}{\partial \hat{y}_i^{(t-1)}} \tag{25}$$

$$h_i = \frac{\partial l\left(\hat{y}_i^{(t-1)}, y_i\right)}{\partial (\hat{y}_i^{(t-1)})^2} \tag{26}$$

where the objective function at the i^{th} iteration, denoted as $L^{(t)}$, is minimized during the training process to produce the best prediction. The

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

previous prediction is represented by $\hat{y}_i^{(t-1)}$, while g_i is the gradient of the loss function with respect to the previous prediction and h_i is the Hessian (second derivative), which measures the sensitivity of the loss function to changes in predictions. The prediction generated by the model, for example x_i, which is the result of the decision tree at iteration t is given by $f_i(x_i)$, which corresponds to the output of the decision tree constructed during that iteration.

4. Calculation of Leaf Weight and Objective Function
For a fixed structure, the optimal weights and objective function can be calculated in Eq. (27) to Eq. (28) as follows [73].

$$w_j = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} \tag{27}$$

$$L^{(t)} = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T \tag{28}$$

where the weight of the j^{th} leaf in the decision tree is denoted as w_j, while the term γT represents the penalty imposed on the number of leaves in the tree. This is part of regularization, which aims to reduce the complexity of the tree.

5. Split Evaluation and Selection
To select the best split, XGBoost calculates the loss reduction after splitting the data at each node. After the split, we have two groups, left I_L and right I_R, where $I = I_R \cup I_L$. The loss reduction after splitting is calculated in Eq. (29) as follows [74]:

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{29}$$

F. Random Search Hyperparameter Tuning

RS is a hyperparameter optimization method that works by defining a probability distribution for each hyperparameter value range, then randomly selecting a combination of values from that distribution to evaluate the model performance [18]. This search process will be stopped when the model performance reaches a certain threshold, or the number of iterations has reached a user-defined limit [75]. RS is known as a simple yet effective approach, as it is able to find high-performance hyperparameter configurations through fewer iterations than methods such as Grid Search, especially in high-dimensional search spaces [76]. Computationally, it is more efficient when dealing with many hyperparameters, as it does not need to evaluate all possible combinations [46].

Another advantage of RS is the flexibility to adjust the search budget to the distribution of the search space. This is particularly useful when some hyperparameters are not evenly distributed, as the randomized approach is more adaptive to such irregularities. Moreover, since each evaluation is performed independently, this method supports parallel execution, thus allowing for optimal utilization of computational resources [77]. With this combination of advantages, RS is a powerful and efficient alternative for hyperparameter optimization, including in complex settings. The performance of the XGBoost model can be improved by tuning the RS Hyperparameter Tuning [78]. XGBoost has a number of important hyperparameters, such as learning rate, maximum number of iterations, and maximum depth [79]. The procedure of RS in Eq. (30) is as follows [80]:

$$Parameter = \arg min_{\theta}\, Loss\ Function\ (\theta) \tag{30}$$

where, θ is a hyperparameter vector to be optimized and $Loss\ Function(\theta)$ is a function that measures the performance of the model based on a particular hyperparameter combination.

RS consists of the following steps [21]:

1. Starting the number of iterations of the parameter combination
2. Initialize all parameter values
3. Repeating a random combination of parameter values based on the number of iterations
4. Performing training using XGBoost on training data
5. Evaluating the resulting classification with test data test data Saving the best value of the classification results and the best combination of parameter values

Table 1. summarizes the hyperparameters and their respective search ranges explored using random search, based on the optimal parameter values reported in previous studies [19], [20], [21], [78].

Table 1. A detailed list of parameters with their respective value ranges in this study.

Parameter	Parameter Value Range
learning_rate	(0.01, ..., 0.1)
max_depth	(1, ..., 12)
n_estimators	(100, ..., 1000)
subsample	(0.6, ..., 1.0)
colsample_bytree	(0.6, ..., 1.0)

G. Evaluation

In this study, model performance was evaluated using a confusion matrix, including accuracy, precision, recall, F1-score, and Area Under the Curve (AUC). The selection of appropriate evaluation metrics is very important to measure the performance of machine learning models objectively and accurately [81].

Metrics such as accuracy, precision, recall, and F1 score, both macro- and weighted-averaged, are used to address the impact of class imbalance. Micro-averaged calculates the total true positives, false positives, and false negatives across all classes, then produces a global metric that gives equal weight to each instance. Macro-averaged calculates the metric for each class separately, then averages without regard to sample size, thus giving equal weight to each class.

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Weighted average, on the other hand, calculates the metric per class then averages with weights based on the sample size in each class, reflecting the unbalanced class distribution in the dataset [82], [83]. The confusion matrix for binary classification is presented in Table. The actual values are coded as True (1) and False (0), while the predicted results are classified as Positive (1) and Negative (0). Four possible classification outcomes, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), are derived from the confusion matrix, with the following explanation [84]:

Table 5. Confusion matrix showing the actual and predicted classifications of the dataset.

Classification		Prediction	
		Positive	Negative
Actual	Positive	True Positive	True Negative
	Negative	False Positive	False Negative

Accuracy, as expressed in Eq. (31) [85], represents the proportion of correct predictions to the total number of predictions made. Mathematically, accuracy is calculated by the following formula [86]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{31}$$

Precision is the ratio between the number of correct positive predictions and the total number of positive predictions. Precision measures the accuracy of positive predictions. The precision, macro precision, and weighted precision are calculated using the formula written in Eq. (32) to Eq. (35) [31] [87] [82]:

$$Precision = \frac{TP}{TP+FP} \tag{32}$$

$$Macro\ Precision = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i} \tag{33}$$

$$Micro\ Precision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i} \tag{34}$$

$$Weighted\ Precision = \frac{1}{\sum_{i=1}^{N}|C_i|}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i} \times |C_i| \tag{35}$$

Recall, also known as sensitivity or True Positive Rate (TPR), is the ratio of correctly predicted positive observations to all observations in the dataset. Recall, macro recall, and weighted recall are calculated using the formula in Eq. (36) to Eq. (39) as follows [31] [87] [82]:

$$Recall = Sensitivity = TPR = \frac{TP}{TP+FN} \tag{36}$$

$$Macro\ Recall = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FN_i} \tag{37}$$

$$Micro\ Recall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FN_i} \tag{38}$$

$$Weighted\ Recall = \frac{1}{\sum_{i=1}^{N}|C_i|}\sum_{i=1}^{N}\frac{TP_i}{TP_i+ FN_i} \times |C_i| \tag{39}$$

The F1-score is a measure of test accuracy. This value is calculated based on precision and recall using the formula in equation (38) [84]. Macro F1 and weighted F1 are calculated using formula in Eq. (40) to Eq. (43) as follows [83] [88] [82]:

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{40}$$

$$Macro\ F1 = \frac{1}{N}\sum_{i=1}^{N}\frac{2\times\frac{TP_i}{TP_i+FP_i}\times\frac{TP_i}{TP_i+FN_i}}{\frac{TP_i}{TP_i+FP_i}+\frac{TP_i}{TP_i+FN_i}} \tag{41}$$

$$Micro\ F1 = \frac{2\times Micro\ Precision \times Micro\ Recall}{Micro\ Precision + Micro\ Recall} \tag{42}$$

$$Weighted\ F1 = \frac{1}{\sum_{i=1}^{N}|C_i|}\sum_{i=1}^{N}\frac{2\times\frac{TP_i}{TP_i+FP_i}\times\frac{TP_i}{TP_i+FN_i}}{\frac{TP_i}{TP_i+FP_i}+\frac{TP_i}{TP_i+FN_i}} \times |C_i| \tag{43}$$

The Area Under the Curve (AUC) is a metric in an ROC diagram that represents the area under the ROC curve of an algorithm. Therefore, the higher the AUC value, the better the performance of the algorithm [85]. The ROC curve depicts the true positive rate (TPR) written in Eq. (36) against the false positive rate (FPR) written in Eq. (44) [31] at various classification thresholds [89]:

$$FPR = \frac{FP}{TN+FP} \tag{44}$$

AUC evaluates a model's ability to distinguish between classes by TPR against FPR; the higher the AUC, the better the model. The higher TPR indicates that the model is performing well. However, accuracy metrics can be misleading when classes are imbalanced, i.e., when one class has more examples. The AUC metric is less affected by class imbalance and provides a comprehensive view of the model's performance across all thresholds [86].

Table 6. Interpretation of the Area Under the Curve (AUC) for model performance evaluation.

Area Under the Curve (AUC)	Interpretation
0,9 <= AUC	Excellent
0,8 <= AUC <= 0,9	Good
0,7 <= AUC <= 0,8	Fair
0,6 <= AUC <= 0,7	Poor
0,5 <= AUC <= 0,6	Fail

AUC stands for “Area Under the ROC Curve”. An ideal ROC curve thus has AUC = 1.0 [16]. For the diagnosis test to be more accurate, the AUC should be greater than 0.5. Generally, an AUC ≥ 0.8 is considered acceptable [90].

III. Result

This section presents the results of cervical cancer classification for four target variables (Hinselmann, Schiller, Cytology, and Biopsy) using XGBoost, XGBoost with Random Search Hyperparameter Tuning with SMOTE, and XGBoost with Random Search Hyperparameter Tuning with ADASYN. The performance of each model is evaluated using accuracy, precision (macro, micro, and weighted),

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

recall (macro, micro, and weighted), F1 score (macro, micro, and weighted), and AUC.

A. XGBoost

In this study, the first classification is using the XGBoost algorithm. Performance was measured using accuracy, precision (macro, micro, and weighted), recall (macro, micro, and weighted), F1-score (macro, micro, and weighted), and AUC. The results of the XGBoost model performance evaluation for the four target variables (Hinselmann, Schiller, Citology, and Biopsy) are presented in Table.

Table 7. Summary of XGBoost model performance measured across various classification evaluation metrics.

Target Variable	Data Split	50:50	60:40	70:30	80:20	90:10
Hinselmann	Accuracy	0.965	0.962	0.961	0.959	0.954
	Precision	0.6	0.6	0.6	0.5	0.5
	Precision Macro	0.787	0.784	0.784	0.735	0.732
	Precision Micro	0.965	0.962	0.961	0.959	0.954
	Precision Weighted	0.959	0.953	0.953	0.951	0.943
	Recall	0.353	0.214	0.273	0.286	0.25
	Recall Macro	0.672	0.604	0.632	0.637	0.619
	Recall Micro	0.965	0.962	0.961	0.959	0.954
	Recall Weighted	0.965	0.962	0.961	0.959	0.954
	F1	0.444	0.316	0.375	0.364	0.333
	F1 Macro	0.713	0.648	0.678	0.671	0.655
	F1 Micro	0.965	0.962	0.961	0.959	0.954
	F1 Weighted	0.961	0.954	0.954	0.954	0.946
	AUC	0.981	0.981	0.986	0.981	0.988
Schiller	Accuracy	0.97	0.968	0.965	0.965	0.954
	Precision	0.9	0.852	0.882	1	0.8
	Precision Macro	0.938	0.915	0.927	0.982	0.882
	Precision Micro	0.97	0.968	0.965	0.965	0.954
	Precision Weighted	0.969	0.967	0.963	0.966	0.95
	Recall	0.73	0.767	0.682	0.6	0.571
	Recall Macro	0.861	0.877	0.837	0.8	0.779
	Recall Micro	0.97	0.968	0.965	0.965	0.954
	Recall Weighted	0.97	0.968	0.965	0.965	0.9535
	F1	0.806	0.807	0.769	0.75	0.667
	F1 Macro	0.895	0.895	0.875	0.866	0.821
	F1 Micro	0.97	0.968	0.965	0.965	0.954
	F1 Weighted	0.968	0.967	0.963	0.961	0.95
	AUC	0.909	0.925	0.86	0.825	0.937
Citology	Accuracy	0.939	0.93	0.93	0.936	0.954
	Precision	0.25	0.125	0.222	0.25	0
	Precision Macro	0.601	0.537	0.589	0.601	0.477
	Precision Micro	0.939	0.93	0.93	0.936	0.954
	Precision Weighted	0.917	0.906	0.919	0.916	0.909
	Recall	0.091	0.056	0.154	0.111	0
	Recall Macro	0.538	0.517	0.563	0.546	0.5
	Recall Micro	0.939	0.93	0.93	0.936	0.954
	Recall Weighted	0.939	0.93	0.93	0.936	0.954
	F1	0.133	0.077	0.182	0.154	0
	F1 Macro	0.551	0.52	0.573	0.56	0.488
	F1 Micro	0.939	0.93	0.93	0.936	0.954
	F1 Weighted	0.926	0.917	0.924	0.924	0.931
	AUC	0.753	0.722	0.641	0.553	0.512
Biopsy	Accuracy	0.944	0.942	0.95	0.959	0.965
	Precision	0.6	0.571	0.625	0.75	0.8
	Precision Macro	0.78	0.765	0.798	0.86	0.887
	Precision Micro	0.944	0.942	0.95	0.959	0.965
	Precision Weighted	0.937	0.933	0.948	0.955	0.963
	Recall	0.429	0.364	0.588	0.546	0.667
	Recall Macro	0.704	0.673	0.782	0.767	0.827
	Recall Micro	0.944	0.942	0.95	0.959	0.965
	Recall Weighted	0.944	0.942	0.95	0.959	0.965

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Table 7. (continued)

Target Variable	Data Split	50:50	60:40	70:30	80:20	90:10
Biopsy	F1	0.5	0.444	0.606	0.632	0.727
	F1 Macro	0.735	0.707	0.79	0.805	0.854
	F1 Micro	0.944	0.942	0.95	0.959	0.965
	F1 Weighted	0.94	0.936	0.949	0.956	0.964
	AUC	0.956	0.923	0.94	0.958	0.973

The best evaluation results of the XGBoost classification on 4 target variables are as follows:

- Hinselmann achieved the best results at a ratio of 50:50 with an accuracy value of 0.965, precision of 0.6, macro precision of 0.787, micro precision of 0.965, weighted precision of 0.959, recall of 0.353, macro recall of 0.672, micro recall of 0.965, weighted recall of 0.965, F1 score of 0.444, macro F1 of 0.713, micro F1 of 0.965, weighted F1 of 0.961, and AUC of 0.981.
- Schiller achieved the best results at a ratio of 50:50 get the best results an accuracy value 0.958, precision 0.788, precision macro 0.88, precision micro 0.958, precision weighted 0.956, recall 0.703, recall macro 0.842, recall micro 0.958, recall weighted 0.958, f1 score 0.743, f1 macro 0.86, f1 micro 0.958, f1 weighted 0.957, and AUC 0.906.
- Citology achieved the best results at a ratio of 90:10, the best results an accuracy value of 0.954, precision 0.5, precision macro 0.732, precision micro 0.954, precision weighted 0.943, recall 0.25, recall macro 0.619, recall micro 0.954, recall weighted 0.954, f1 score 0.333, f1 macro 0.655, f1 micro 0.954, f1 weighted 0.946, and AUC 0.537.
- Biopsy achieved the best results at a ratio of 90:10, the best results an accuracy value of 0.965, precision 0.8, precision macro 0.888, precision micro 0.965, precision weighted 0.963, recall 0.667, recall macro 0.827, recall micro 0.965, recall weighted 0.965, f1 score 0.727, f1 macro 0.854, f1 micro 0.965, f1 weighted 0.964, and AUC 0.96.

B. XGBoost with RSHT with SMOTE

The second classification in this study utilised the XGBoost algorithm optimised using Random Search Hyperparameter Tuning (RSHT), and the SMOTE technique to handle imbalanced data. The results of the SMOTE technique implementation are presented in Table . The best parameters from the Random Search results for XGBoost are presented in Table 9.. The overall classification results for the four target variables (Hinselmann, Schiller, Cytology, and Biopsy) using the XGBoost method with a combination of RSHT and SMOTE are presented sequentially in Table. Based on Table 8. and Fig., before the application of SMOTE, the number of samples for the minority class (1) was less than the majority class (0). After SMOTE, the number of samples in the minority class was successfully increased, resulting in balanced proportions with the majority class across all target variables. The target variables on the X-axis represent different diagnostic tests: Hinselmann, Schiller, Cytology, and Biopsy, while the Y-axis shows the total number of samples (count) in the dataset. For example, the Hinselmann minority class increased from 18 to 411, the Schiller class from 37 to 392, the cytology class from 40 to 732, and the biopsy class from 44 to 642.

Based on Table 9., the best parameter values for random search on 4 target variables are as follows:

- Hinselmann achieved the best parameter at a ratio 50:50, the best parameter is learning_rate 0.051, max_depth 1, n_estimators 929, subsample 0.989, and colsample_bytree 0.808
- Schiller achieved the best parameter at a ratio of 50:50. The best parameters are learning_rate

Table 8. Comparison of dataset distributions before and after applying the SMOTE oversampling technique.

Target Variable	Data Split	Before SMOTE		After SMOTE	
		0	1	0	1
Hinselmann	50:50	411	18	411	411
	60:40	493	21	493	493
	70:30	576	24	576	576
	80:20	658	28	658	658
	90:10	741	31	741	741
Schiller	50:50	392	37	392	392
	60:40	470	44	470	470
	70:30	548	52	548	548
	80:20	627	59	627	627
	90:10	705	67	705	705
Citology	50:50	407	22	407	407
	60:40	488	26	488	488
	70:30	569	31	569	569
	80:20	651	35	651	651
	90:10	732	40	732	732
Biopsy	50:50	402	27	402	402
	60:40	481	33	481	481
	70:30	562	38	562	562
	80:20	642	44	642	642
	90:10	723	49	723	723

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

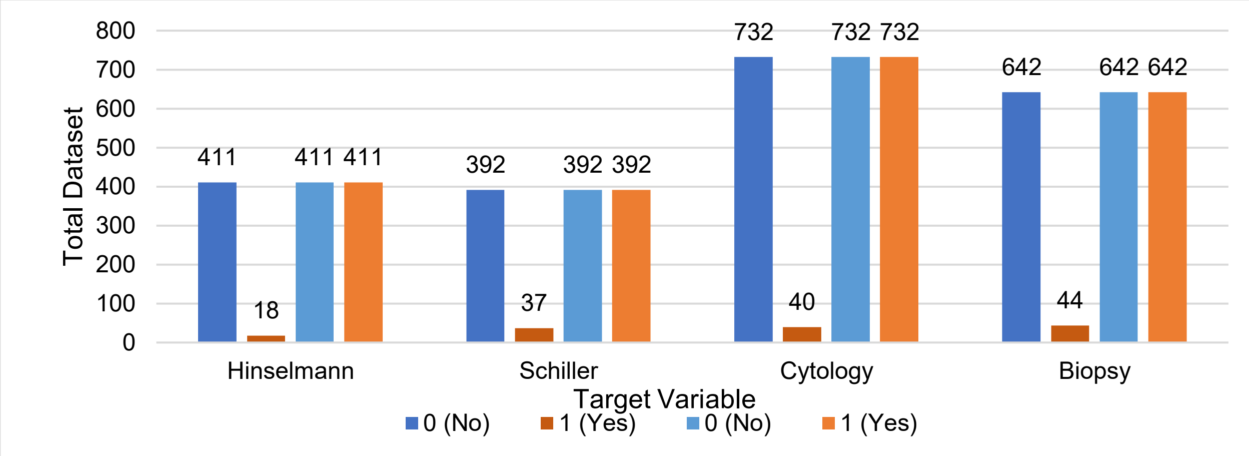

Fig. 2. Visualization of the dataset showing differences before and after SMOTE oversampling applied.

Table 9. Table presenting the hyperparameter setup for Random Search with corresponding value ranges tested.

Target variable	Data Split	Parameter				
		learning_rate	max_depth	n_estimators	subsample	colsample_b
Hinselmann	50:50	0.051	1	929	0.989	0.808
	60:40	0.019	9	256	0.921	0.743
	70:30	0.097	8	228	0.68	0.63
	80:20	0.046	2	995	0.722	0.641
	90:10	0.081	4	661	0.865	0.614
Schiller	50:50	0.079	4	353	0.604	0.665
	60:40	0.081	4	661	0.865	0.614
	70:30	0.044	9	727	0.741	0.709
	80:20	0.014	11	558	0.947	0.662
	90:10	0.089	11	171	0.84	0.75
Citology	50:50	0.014	11	558	0.947	0.662
	60:40	0.079	4	353	0.604	0.665
	70:30	0.097	8	228	0.68	0.63
	80:20	0.084	10	710	0.691	0.646
	90:10	0.096	6	610	0.636	0.788
Biopsy	50:50	0.037	4	833	0.656	0.79
	60:40	0.019	9	256	0.921	0.743
	70:30	0.014	11	558	0.947	0.662
	80:20	0.014	11	558	0.947	0.662
	90:10	0.084	8	561	0.678	0.839

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Table 10. Evaluation of XGBoost performance using random Search Hyperparameter Tuning and SMOTE.

Target Variable	Data Split	50:50	60:40	70:30	80:20	90:10
Hinselmann	Accuracy	0.967	0.962	0.954	0.959	0.954
	Precision	0.615	0.556	0.4	0.5	0.5
	Precision Macro	0.797	0.764	0.682	0.735	0.732
	Precision Micro	0.967	0.962	0.954	0.959	0.954
	Precision Weighted	0.964	0.956	0.94	0.951	0.943
	Recall	0.471	0.357	0.182	0.286	0.25
	Recall Macro	0.729	0.673	0.585	0.637	0.619
	Recall Micro	0.967	0.962	0.954	0.959	0.954
	Recall Weighted	0.967	0.962	0.954	0.959	0.954
	F1	0.533	0.435	0.25	0.364	0.333
	F1 Macro	0.758	0.708	0.613	0.671	0.655
	F1 Micro	0.967	0.962	0.954	0.959	0.954
	F1 Weighted	0.965	0.958	0.945	0.954	0.946
	AUC	0.943	0.981	0.979	0.985	0.988
Schiller	Accuracy	0.958	0.968	0.961	0.959	0.942
	Precision	0.788	0.88	0.833	0.9	0.667
	Precision Macro	0.88	0.928	0.902	0.932	0.815
	Precision Micro	0.958	0.968	0.961	0.959	0.942
	Precision Weighted	0.956	0.967	0.959	0.958	0.938
	Recall	0.703	0.733	0.682	0.6	0.571
	Recall Macro	0.842	0.862	0.835	0.797	0.773
	Recall Micro	0.958	0.968	0.961	0.959	0.942
	Recall Weighted	0.958	0.968	0.961	0.959	0.942
	F1	0.743	0.8	0.75	0.72	0.615
	F1 Macro	0.86	0.891	0.865	0.849	0.792
	F1 Micro	0.958	0.968	0.961	0.959	0.942
	F1 Weighted	0.957	0.967	0.96	0.956	0.94
	AUC	0.906	0.92	0.865	0.859	0.81
Cytology	Accuracy	0.939	0.927	0.93	0.924	0.954
	Precision	0.333	0.182	0.273	0.25	0.5
	Precision Macro	0.645	0.567	0.616	0.604	0.732
	Precision Micro	0.939	0.927	0.93	0.924	0.954
	Precision Weighted	0.925	0.912	0.925	0.92	0.943
	Recall	0.182	0.111	0.231	0.222	0.25
	Recall Macro	0.581	0.542	0.599	0.593	0.619
	Recall Micro	0.939	0.927	0.93	0.924	0.954
	Recall Weighted	0.939	0.927	0.93	0.924	0.954
	F1	0.235	0.138	0.25	0.235	0.333
	F1 Macro	0.602	0.55	0.607	0.598	0.655
	F1 Micro	0.939	0.927	0.93	0.924	0.954
	F1 Weighted	0.931	0.919	0.928	0.922	0.946
	AUC	0.735	0.698	0.644	0.552	0.537
Biopsy	Accuracy	0.935	0.954	0.938	0.971	0.965
	Precision	0.5	0.625	0.533	0.8	0.8
	Precision Macro	0.734	0.802	0.748	0.891	0.888
	Precision Micro	0.935	0.802	0.938	0.971	0.965
	Precision Weighted	0.936	0.956	0.935	0.97	0.963
	Recall	0.536	0.682	0.471	0.727	0.667
	Recall Macro	0.749	0.827	0.721	0.857	0.827
	Recall Micro	0.935	0.954	0.938	0.971	0.965
	Recall Weighted	0.935	0.954	0.938	0.971	0.965
	F1	0.517	0.652	0.5	0.762	0.727
	F1 Macro	0.741	0.814	0.734	0.873	0.854
	F1 Micro	0.935	0.954	0.938	0.971	0.965
	F1 Weighted	0.936	0.954	0.936	0.97	0.964
	AUC	0.921	0.933	0.892	0.956	0.96

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

0.0792, max_depth 4, n_estimators 353, subsample 0.604, and colsample_bytree 0.665.

- Citology achieved the best parameter at a ratio 90:10. The best parameters are learning_rate 0.096, max_depth 6, n_estimators 610, subsample 0.636, and colsample_bytree 0.788
- Biopsy achieved the best parameter at a ratio of 80:20. The best parameters are learning_rate 0.014, max_depth 11, n_estimators 558, subsample 0.947, and colsample_bytree 0.662.

The best evaluation results from classification using a combination of XGBoost with RSHT and SMOTE on four target variables are as follows:

- Hinselmann achieved the best results at a ratio of 50:50 get the best results an accuracy value 0.967, precision 0.615, precision macro 0.797, precision micro 0.967, precision weighted 0.964, recall 0.471, recall macro 0.729, recall micro 0.967, recall weighted 0.967, F1 score 0.533, F1 macro 0.758, F1 micro 0.967, F1 weighted 0.965, and AUC 0.943.
- Schiller achieved the best results at a ratio of 60:40, the best result an accuracy value of 0.968, precision 0.88, precision macro 0.928, precision micro 0.968, precision weighted 0.966, recall 0.733, recall macro 0.862, recall micro 0.968, recall weighted 0.968, F1 score 0.8, F1 macro 0.891, F1 micro 0.968, F1 weighted 0.967, and AUC 0.92.
- Citology achieved the best results at a ratio of 90:10, the best results an accuracy value of 0.954, precision 0.5, precision macro 0.732, precision micro 0.954, precision weighted 0.943, recall 0.25, recall macro 0.619, recall micro 0.954, recall weighted 0.954, F1 score 0.333, F1 macro 0.655, F1 micro 0.954, F1 weighted 0.946, and AUC 0.537.

Table 11. Comparison of dataset distributions before and after applying the ADASYN oversampling technique.

Target Variable	Data Split	Before ADASYN		After ADASYN	
		0	1	0	1
Hinselmann	50:50	411	18	411	417
	60:40	493	21	493	492
	70:30	576	24	576	568
	80:20	658	28	658	657
	90:10	741	31	741	736
Schiller	50:50	392	37	392	392
	60:40	470	44	470	467
	70:30	548	52	548	550
	80:20	627	59	627	622
	90:10	705	67	705	728
Citology	50:50	407	22	407	414
	60:40	488	26	488	492
	70:30	569	31	569	576
	80:20	651	35	651	640
	90:10	732	40	732	727
Biopsy	50:50	402	27	402	396
	60:40	481	33	481	477
	70:30	562	38	562	578
	80:20	642	44	642	662
	90:10	723	49	723	706

- Biopsy achieved the best results at a ratio of 80:20, the best results an accuracy value of 0.971, precision 0.8, precision macro 0.891, precision micro 0.971, precision weighted 0.97, recall 0.727, recall macro 0.857, recall micro 0.971, recall weighted 0.971, F1 score 0.761, F1

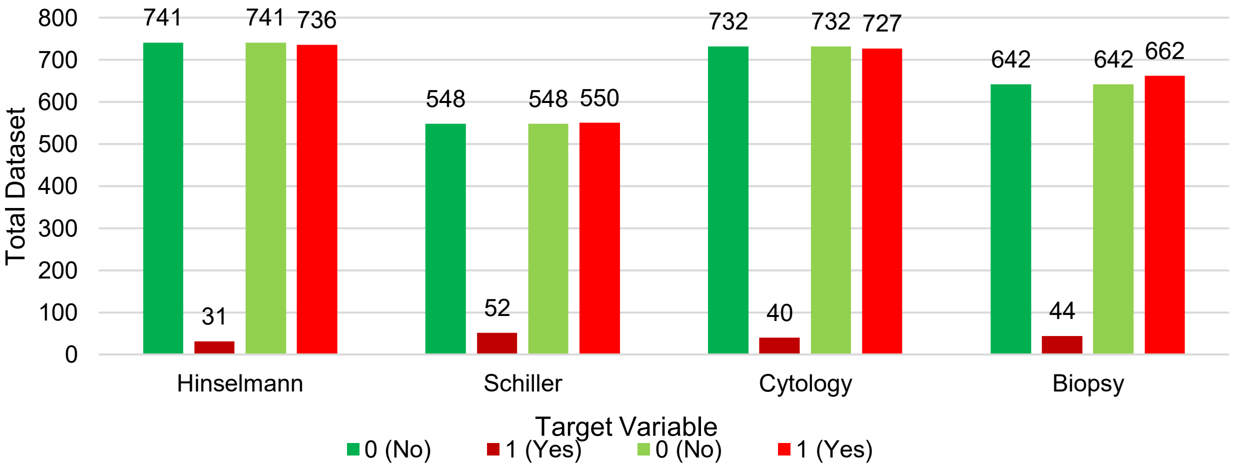

Fig. 3. Visualization of the dataset showing differences before and after ADASYN oversampling applied.

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

macro 0.873, F1 micro 0.971, F1 weighted 0.97, and AUC 0.956.

C. XGBoost with RSHT with ADASYN

The third classification using the XGBoost algorithm optimised with Random Search HT, and the ADASYN technique to handle imbalanced data. The results of implementing the ADASYN technique are provided in Table 11. The best parameters from the Random Search for XGBoost are described in Table 12. The classification results for the four target variables (Hinselmann, Schiller, Cytology, Biopsy) using the XGBoost method with a combination of RSHT and ADASYN are presented in Table 13. Based on Table11 and Fig. 3 before the application of ADASYN, the number of samples for the minority class (1) was less than that for the majority class (0). After ADASYN, the number of samples for the minority class was successfully enhanced to balance with the majority class on all target variables. The target variables on the X-axis represent different diagnostic tests: Hinselmann, Schiller, Cytology, and Biopsy, while the Y-axis shows the total number of samples (count) in the dataset. For example, the Hinselmann minority class increased from 31 to 736, the Schiller class from 52 to 550, the Cytology class from 40 to 727, and the Biopsy class from 44 to 662.

Table 12. Table presenting the hyperparameter setup for Random Search with corresponding value ranges tested.

Target variable	Data Split	Parameter				
		learning_rate	max_depth	n_estimators	subsample	colsample_bytree
Hinselmann	50:50	0.021	2	978	0.932	0.618
	60:40	0.046	2	995	0.722	0.641
	70:30	0.021	2	978	0.932	0.618
	80:20	0.089	2	876	0.606	0.626
	90:10	0.046	2	995	0.722	0.641
Schiller	50:50	0.096	4	671	0.704	0.67
	60:40	0.031	6	515	0.837	0.658
	70:30	0.014	11	558	0.947	0.662
	80:20	0.084	10	710	0.691	0.645
	90:10	0.084	10	710	0.691	0.645
Citology	50:50	0.081	4	661	0.865	0.614
	60:40	0.019	8	926	0.729	0.877
	70:30	0.084	10	710	0.691	0.645
	80:20	0.084	10	710	0.691	0.645
	90:10	0.096	6	610	0.636	0.788
Biopsy	50:50	0.044	9	727	0.741	0.709
	60:40	0.074	4	897	0.953	0.853
	70:30	0.089	2	876	0.606	0.626
	80:20	0.081	4	661	0.865	0.614
	90:10	0.046	8	825	0.955	0.867

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Table 13. **Evaluation of XGBoost performance using random Search Hyperparameter Tuning and ADASYN.**

Target Variable	Data Split	50:50	60:40	70:30	80:20	90:10
Hinselmann	Accuracy	0.965	0.956	0.957	0.959	0.965
	Precision	0.583	0.429	0.5	0.5	0.667
	Precision Macro	0.78	0.698	0.734	0.735	0.821
	Precision Micro	0.965	0.956	0.957	0.959	0.965
	Precision Weighted	0.961	0.945	0.948	0.951	0.962
	Recall	0.412	0.214	0.273	0.286	0.5
	Recall Macro	0.7	0.601	0.63	0.637	0.744
	Recall Micro	0.965	0.956	0.957	0.959	0.965
	Recall Weighted	0.965	0.956	0.957	0.959	0.965
	F1	0.483	0.286	0.353	0.364	0.571
	F1 Macro	0.732	0.632	0.665	0.671	0.777
	F1 Micro	0.965	0.956	0.957	0.959	0.965
	F1 Weighted	0.962	0.949	0.951	0.954	0.963
	AUC	0.963	0.954	0.95	0.985	0.988
Schiller	Accuracy	0.96	0.965	0.969	0.959	0.954
	Precision	0.794	0.846	0.937	0.9	0.8
	Precision Macro	0.884	0.911	0.954	0.932	0.882
	Precision Micro	0.96	0.965	0.969	0.959	0.954
	Precision Weighted	0.959	0.964	0.968	0.958	0.95
	Recall	0.73	0.733	0.682	0.6	0.571
	Recall Macro	0.856	0.86	0.839	0.797	0.779
	Recall Micro	0.96	0.965	0.969	0.959	0.954
	Recall Weighted	0.96	0.965	0.969	0.959	0.954
	F1	0.761	0.786	0.79	0.72	0.667
	F1 Macro	0.87	0.883	0.886	0.849	0.821
	F1 Micro	0.96	0.965	0.969	0.959	0.954
	F1 Weighted	0.96	0.964	0.967	0.956	0.95
	AUC	0.914	0.9159	0.892	0.871	0.863
Citology	Accuracy	0.932	0.924	0.923	0.901	0.954
	Precision	0.182	0.1	0.231	0.1	0
	Precision Macro	0.567	0.525	0.595	0.525	0.477
	Precision Micro	0.932	0.924	0.923	0.901	0.954
	Precision Weighted	0.913	0.905	0.923	0.906	0.909
	Recall	0.091	0.056	0.231	0.111	0
	Recall Macro	0.534	0.514	0.595	0.528	0.5
	Recall Micro	0.9324	0.9244	0.9225	0.9012	0.9535
	Recall Weighted	0.932	0.924	0.923	0.901	0.954
	F1	0.121	0.071	0.231	0.105	0
	F1 Macro	0.543	0.516	0.595	0.527	0.488
	F1 Micro	0.932	0.924	0.923	0.901	0.954
	F1 Weighted	0.922	0.914	0.923	0.904	0.931
	AUC	0.68	0.719	0.631	0.556	0.555
Biopsy	Accuracy	0.939	0.933	0.938	0.971	0.954
	Precision	0.531	0.467	0.533	0.8	0.75
	Precision Macro	0.752	0.711	0.748	0.891	0.857
	Precision Micro	0.939	0.933	0.938	0.971	0.954
	Precision Weighted	0.944	0.923	0.935	0.97	0.949
	Recall	0.607	0.318	0.471	0.727	0.5
	Recall Macro	0.784	0.647	0.721	0.857	0.744
	Recall Micro	0.939	0.933	0.938	0.971	0.954
	Recall Weighted	0.939	0.933	0.938	0.971	0.954
	F1	0.567	0.378	0.5	0.762	0.6
	F1 Macro	0.767	0.672	0.734	0.873	0.788
	F1 Micro	0.939	0.933	0.938	0.971	0.954
	F1 Weighted	0.941	0.927	0.936	0.97	0.949
	AUC	0.928	0.885	0.892	0.956	0.971

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Based on Table, the best parameter values for Random search on 4 target variables is as follows:

- Hinselmann achieved the best parameter at a ratio of 90:10. The best parameters are learning_rate 0.046, max_depth 2, n_estimators 995, subsample 0.722, and colsample_bytree 0.641.
- Schiller achieved the best parameter at a ratio of 70:30. The best parameters are learning_rate 0.014, max_depth 11, n_estimators 558, subsample 0.947, and colsample_bytree 0.662.
- Citology achieved the best parameter at a ratio of 90:10. The best parameters are learning_rate 0.096, max_depth 6, n_estimators 610, subsample 0.636, and colsample_bytree 0.788.
- Biopsy achieved the best parameter at a ratio of 80:20. The best parameters are learning_rate 0.081, max_depth 4, n_estimators 661, subsample 0.865, and colsample_bytree 0.614.

The best evaluation results from classification using a combination of XGBoost with RSHT and ADASYN on four target variables are as follows, first at Hinselmann achieved the best results at a ratio of 90:10, the best results an accuracy value of 0.965, precision 0.667, precision macro 0.821, precision micro 0.965, precision weighted 0.962, recall 0.5, recall macro 0.744, recall micro 0.965, recall weighted 0.965, F1 score 0.571, F1 macro 0.777, F1 micro 0.965, F1 weighted 0.963, and AUC 0.988. Second, at Schiller achieved the best results at a ratio of 70:30, the best results an accuracy value of 0.969, precision 0.938, precision macro 0.954, precision micro 0.969, precision weighted 0.968, recall 0.682, recall macro 0.839, recall micro 0.969, recall weighted 0.969, F1 score 0.79, F1 macro 0.886, F1 micro 0.969, F1 weighted 0.967, and AUC 0.892. Third, at Citology achieved the best results at a ratio of 90:10, the best results an accuracy value of 0.954, precision 0, precision macro 0.477, precision micro 0.954, precision weighted 0.909, recall 0, recall macro 0.5, recall micro 0.954, recall weighted 0.954, F1 score 0, F1 macro 0.488, F1 micro 0.954, F1 weighted 0.931, and AUC 0.555. The last, for Biopsy achieved the best results at a ratio of 80:20, the best results an accuracy value of 0.971, precision 0.8, precision macro 0.891, precision micro 0.971, precision weighted 0.97, recall 0.727, recall macro 0.857, recall micro 0.971, recall weighted 0.971, F1 score 0.762, F1 macro 0.873, F1 micro 0.971, F1 weighted 0.97, and AUC 0.956.

IV. Discussion

Drawing on the research outcomes outlined earlier, the top-performing results emerged from three models: the standard XGBoost, XGBoost enhanced with Random Search Hyperparameter Tuning (RSHT) and SMOTE, and XGBoost integrated with RSHT and ADASYN. A detailed comparison of the optimal performance across these models is illustrated in Table. As indicated in Table., incorporating optimization and data balancing strategies markedly boosted the XGBoost model's effectiveness. The basic XGBoost model, without RSHT, SMOTE, or ADASYN, exhibited the lowest accuracy and AUC scores across the three experimental setups. In contrast, XGBoost models augmented with RSHT alongside SMOTE or ADASYN delivered substantially improved outcomes. Notably, the XGBoost with RSHT and SMOTE configuration consistently secured the top accuracy and AUC values across all target variables,

Table 14. Comparison of the best performance of all classification models in this study.

Target Variable		Hinselmann	Schiller	Citology	Biopsy
XGB	Accuracy	0.965	0.97	0.954	0.965
	Precision	0.6	0.9	0	0.8
	Precision Macro	0.7869	0.938	0.477	0.888
	Precision Micro	0.965	0.97	0.954	0.965
	Precision Weighted	0.959	0.969	0.909	0.963
	Recall	0.353	0.73	0	0.667
	Recall Macro	0.672	0.861	0.5	0.827
	Recall Micro	0.965	0.97	0.954	0.965
	Recall Weighted	0.965	0.97	0.954	0.965
	F1	0.444	0.806	0	0.727
	F1 Macro	0.713	0.895	0.488	0.854
	F1 Micro	0.965	0.97	0.954	0.965
	F1 Weighted	0.961	0.968	0.931	0.964
	AUC	0.981	0.909	0.512	0.973
XGB + RSHT + SMOTE	Accuracy	0.967	0.968	0.954	0.971
	Precision	0.615	0.88	0.5	0.8
	Precision Macro	0.797	0.928	0.732	0.891
	Precision Micro	0.967	0.968	0.954	0.971
	Precision Weighted	0.964	0.966	0.943	0.97
	Recall	0.471	0.733	0.25	0.727
	Recall Macro	0.729	0.862	0.619	0.857
	Recall Micro	0.967	0.968	0.954	0.971

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Table 14. (continued)

Target Variable		Hinselmann	Schiller	Citology	Biopsy
XGB + RSHT + SMOTE	Recall Weighted	0.967	0.968	0.954	0.971
	F1	0.533	0.8	0.333	0.762
	F1 Macro	0.758	0.891	0.655	0.873
	F1 Micro	0.967	0.968	0.954	0.971
	F1 Weighted	0.965	0.967	0.946	0.971
	AUC	0.943	0.92	0.537	0.956
XGB + RSHT + ADASYN	Accuracy	0.965	0.969	0.954	0.971
	Precision	0.667	0.938	0	0.8
	Precision Macro	0.821	0.954	0.477	0.891
	Precision Micro	0.965	0.969	0.954	0.971
	Precision Weighted	0.962	0.968	0.909	0.97
	Recall	0.5	0.682	0	0.727
	Recall Macro	0.744	0.839	0.5	0.857
	Recall Micro	0.965	0.969	0.954	0.971
	Recall Weighted	0.965	0.969	0.954	0.971
	F1	0.571	0.789	0	0.762
	F1 Macro	0.777	0.886	0.488	0.873
	F1 Micro	0.965	0.969	0.954	0.971
	F1 Weighted	0.963	0.967	0.931	0.97
	AUC	0.988	0.892	0.555	0.956

Table 15. Comparison with previous research on cervical cancer classification results in this study

Research	Target Variable	Classifier	Accuracy
[91]	Biopsy	Random Forest + SMOTE	0.95
		SVM + SMOTE	0.91
		Decision Tree + SMOTE	0.95
[92]	Biopsy	XGB + KNN Imputer	0.835
		XGB + Deleted Missing Value	0.734
[82]	Biopsy	KNN + SMOTE	0.884
		KNN + ADASYN	0.879
Proposed Work	Hinselmann	XGBoost	0.965
		XGBoost + RSHT + SMOTE	0.967
		XGBoost + RSHT + ADASYN	0.965
	Schiller	XGBoost	0.97
		XGBoost + RSHT + SMOTE	0.968
		XGBoost + RSHT + ADASYN	0.969
	Citology	XGBoost	0.954
		XGBoost + RSHT + SMOTE	0.954
		XGBoost + RSHT + ADASYN	0.954
	Biopsy	XGBoost	0.965
		XGBoost + RSHT + SMOTE	0.971
		XGBoost + RSHT + ADASYN	0.971

including Hinselmann, Schiller, Cytology, and Biopsy. While ADASYN also yielded strong results, SMOTE's edge was clear in its reliability across recall and F1-score metrics, which are more significant in medical diagnostics due to their focus on thorough detection of positive instances. This suggests that ADASYN's gains in precision fall short of matching SMOTE's steadiness in these vital areas. These findings affirm that fine-tuning parameters via RSHT and addressing data imbalances with SMOTE are essential for optimizing classification model performance, representing the most potent setup for this dataset. An earlier investigation referenced as [91] employed the 'Cervical Cancer Risk Factors' dataset to identify cervical

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (DOI): https://doi.org/10.35882/jeeemi.v8i1.1415
Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

cancer, focusing on biopsy as the outcome variable. This was achieved using multiple machines learning techniques, including Random Forests, SVMs, and Decision Trees, and integratingSMOTE for data balancing. Following that, the work in [92] leveraged the same dataset to pinpoint cervical cancer cases, again targeting biopsy, by assessing the XGBoost algorithm alongside various imputation strategies like KNN Imputer and the removal of missing values. Additionally, the research from [82] applied the KNN approach to detect cervical cancer, with biopsy as the key variable, combined with distinct data balancing techniques such as SMOTE and ADASYN. A summary of how these prior studies compare with the present research is outlined in Table. Based on the experimental results and the comparison with previous studies presented in Table., the proposed framework demonstrates clear and consistent performance advantages in cervical cancer classification. Prior research generally agrees that ensemble classifiers combined with data balancing techniques improve predictive accuracy, particularly when handling imbalanced medical datasets. Studies employing SMOTE or ADASYN with conventional classifiers and XGBoost report notable performance gains; however, variations in accuracy remain largely influenced by preprocessing strategies and dataset characteristics.

In this context, the proposed approach combining XGBoost, Random Search Hyperparameter Tuning (RSHT), and SMOTE consistently outperforms XGBoost with RSHT and ADASYN, as well as methods reported in previous studies, across all four diagnostic targets (Hinselmann, Schiller, Cytology, and Biopsy). This superiority is reflected in higher accuracy, AUC, and precision, recall, and F1-score under macro, micro, and weighted schemes, with micro-averaged metrics showing the strongest performance, indicating robust overall predictive capability in multitarget cervical cancer diagnosis. The observed advantage of SMOTE over ADASYN can be attributed to its ability to generate evenly distributed synthetic samples for the underrepresented class, whereas ADASYN emphasizes hard-to-learn samples, which can introduce noise and inconsistencies in datasets with multiple targets, such as Cytology. This difference likely contributes to the more stable and higher performance achieved with SMOTE.

The training–testing data ratio also affects performance. Splits with larger training sets, particularly 70:30 and 80:20, generally produce more stable and higher results, as the model can better learn complex patterns following RSHT optimization and SMOTE balancing. Conversely, smaller training sets (50:50) limit learning capacity, while very small test sets (90:10) increase evaluation variance. Overall, an 80:20 split offers a good balance between model robustness and evaluation reliability. Performance varies across the four diagnostic target variables. While Hinselmann and Biopsy consistently achieve higher AUC values, the Cytology target exhibits substantially lower discriminative performance, with AUC values approaching random classification (0.5) for certain data splits, such as 80:20 and 90:10. This reduced performance is likely attributable to the subjective nature of cytological assessment and the presence of ambiguous class boundaries, which increase label variability and limit model separability. In contrast, Hinselmann and Biopsy rely on more objective visual or histopathological evidence, resulting in more consistent annotations and stronger discriminative signals. From a clinical perspective, these findings suggest that Cytology-based predictions should be interpreted with caution, while from a methodological standpoint, they highlight the need for target-specific modeling strategies or complementary features to improve performance for diagnostically challenging outcomes. The observed improvement is attributable to the combination of implemented techniques. XGBoost iteratively builds strong classifiers, RSHT optimizes parameters systematically, SMOTE mitigates class imbalance by generating synthetic minority samples, and MICE handles missing values to preserve feature relationships, collectively enhancing model accuracy and clinical relevance.

Despite the performance improvements achieved, several limitations should be acknowledged. Although ADASYN effectively mitigates class imbalance, it may generate noisy or unevenly distributed synthetic samples, potentially affecting model stability and generalization, particularly in terms of accuracy and AUC. Specifically, an increase in recall was accompanied by a decrease in AUC across several target variables, suggesting that the synthetic samples generated by ADASYN may have introduced noise or increased class overlap. In addition, relying on confusion matrix-based metrics without incorporating macro, micro, and weighted averaging may result in biased performance interpretations in imbalanced and multitarget settings, as such evaluations are often dominated by the majority classes. Further research should employ larger and more diverse datasets, alternative balancing methods, and varied model configurations to improve robustness and clinical relevance.

V. Conclusion

This study aimed to comparatively evaluate SMOTE and ADASYN for addressing class imbalance in cervical cancer identification using an XGBoost classifier optimized through Random Search Hyperparameter Tuning (RSHT) and supported by Multiple Imputation by Chained Equations (MICE). Four diagnostic outcomes, Hinselmann, Schiller, Cytology,

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415
Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

and Biopsy, were treated as independent binary classification tasks. The results demonstrate that integrating XGBoost, RSHT, MICE, and SMOTE delivers superior overall performance. The proposed framework achieved 97.1% accuracy, micro-precision, micro-recall, micro-F1 score, and AUC, confirming its robustness and effectiveness in multitarget cervical cancer classification. In contrast, the ADASYN-based configuration produced marginally lower micro-averaged performance and substantially reduced AUC, indicating less reliable class separability under imbalanced conditions. Further analysis at the target level shows that the SMOTE-based model maintains consistently high accuracy across all diagnostic outcomes, with performance ranging from approximately 95.4% to 97.1%, where Biopsy and Hinselmann exhibit stronger discriminative capability than Cytology. These variations highlight the influence of target-specific characteristics on classification performance, particularly in diagnostically challenging outcomes. Future work should explore larger and more diverse datasets, hybrid imbalance-handling strategies, and alternative ensemble orthat deep learning architectures to further enhance robustness and clinical applicability. Emphasis should also be placed on developing target-aware modeling strategies and evaluating deployment feasibility in real-world clinical screening environments. Overall, this study confirms that combining MICE imputation, SMOTE oversampling, and optimized XGBoost provide a reliable and practical framework for cervical cancer detection in imbalanced multitarget datasets.

Acknowledgment

The authors extend their profound appreciation to the Computer Science Study Program at the Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University, for the sustained support and resources provided throughout this study. The nurturing academic environment, combined with the insightful mentorship and motivation provided by the faculty, proved essential to the successful realization of this research endeavor.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The dataset used in this study is the Cervical Cancer (Risk Factors) Dataset from the University of California, Irvine (UCI) Machine Learning Repository, which contains patient data from “Hospital Universitario de Caracas” in Caracas, Venezuela. This dataset can be accessed through the following link: https://archive.ics.uci.edu/dataset/383/cervical+cancer+risk+factors.

Author Contribution

Mita Azzahra Ramadhan served as the lead author, responsible for formulating research ideas, designing the research framework, collecting data, and analyzing and interpreting the research results. Triando Hamonangan Saragih provided substantive guidance in developing the research methodology and conceptual framework. Dwi Kartini contributed to technical assistance and the development of the research implementation aspects. Muhammad Itqan Mazdadi contribute to the evaluation and testing of research results and provided input to improve the quality and reliability of the findings. Muliadi also provided critical reviews of the manuscript and input for improving the writing. All authors contributed to the final review of the manuscript and approved the final version for publication.

Declarations

Ethical Approval

The present study was undertaken to meet the requirements of the student's final project and adheres to all academic regulations and guidelines established by the Computer Science Study Program at the Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University.

Consent for Publication Participants.

Consent for publication was given by all participants

Competing Interests

The authors declare no competing interests.

References

[1] L. W. Habtemariam, E. T. Zewde, and G. L. Simegn, “Cervix Type and Cervical Cancer Classification System Using Deep Learning Techniques,” *Medical Devices: Evidence and Research*, vol. 15, pp. 163–176, 2022, doi: 10.2147/MDER.S366303.

[2] J. J. Tanimu, M. Hamada, M. Hassan, H. A. Kakudi, and J. O. Abiodun, “A Machine Learning Method for Classification of Cervical Cancer,” *Electronics (Switzerland)*, vol. 11, no. 3, Feb. 2022, doi: 10.3390/electronics11030463.

[3] S. Umirzakova, S. Muksimova, J. Baltayev, and Y. I. Cho, “Force Map-Enhanced Segmentation of a Lightweight Model for the Early Detection of Cervical Cancer,” *Diagnostics*, vol. 15, no. 5, p. 513, Feb. 2025, doi: 10.3390/diagnostics15050513.

[4] K. Wdowiak, A. Drab, P. Filipek, and U. Religioni, “The Assessment of Knowledge About Cervical Cancer, HPV Vaccinations, and Screening

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415
Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Programs Among Women as an Element of Cervical Cancer Prevention in Poland," *J Pers Med*, vol. 14, no. 12, p. 1139, Dec. 2024, doi: 10.3390/jpm14121139.

[5] N. Bhatla, D. Aoki, D. N. Sharma, and R. Sankaranarayanan, "Cancer of the cervix uteri: 2021 update," *International Journal of Gynecology and Obstetrics*, vol. 155, no. S1, pp. 28–44, Oct. 2021, doi: 10.1002/ijgo.13865.

[6] C. K. Maswanganye, P. P. Mkhize, and N. D. Matume, "Mapping the HPV Landscape in South African Women: A Systematic Review and Meta-Analysis of Viral Genotypes, Microbiota, and Immune Signals," Dec. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/v16121893.

[7] C. Buelens *et al.*, "Experiences and Perceptions of Cervical Cancer Screening Using Self-Sampling among Under-Screened Women in Flanders," *Healthcare*, vol. 12, no. 17, p. 1704, Aug. 2024, doi: 10.3390/healthcare12171704.

[8] A. Sukhamwang, S. Inthanon, P. Dejkriengkraikul, T. Semangoen, and S. Yodkeeree, "Anti-Cancer Potential of Isoflavone-Enriched Fraction from Traditional Thai Fermented Soybean against Hela Cervical Cancer Cells," *Int J Mol Sci*, vol. 25, no. 17, p. 9277, Aug. 2024, doi: 10.3390/ijms25179277.

[9] P. E. Castle, "Looking Back, Moving Forward: Challenges and Opportunities for Global Cervical Cancer Prevention and Control," *Viruses*, vol. 16, no. 9, p. 1357, Aug. 2024, doi: 10.3390/v16091357.

[10] N. Al Mudawi and A. Alazeb, "A Model for Predicting Cervical Cancer Using Machine Learning Algorithms," *Sensors*, vol. 22, no. 11, Jun. 2022, doi: 10.3390/s22114132.

[11] S. Gil-Rojas *et al.*, "Application of Machine Learning Techniques to Assess Alpha-Fetoprotein at Diagnosis of Hepatocellular Carcinoma," *Int J Mol Sci*, vol. 25, no. 4, Feb. 2024, doi: 10.3390/ijms25041996.

[12] Z. Liu, S. Zhang, H. Zhang, and X. Li, "A Study on Caregiver Activity Recognition for the Elderly at Home Based on the XGBoost Model," *Mathematics*, vol. 12, no. 11, Jun. 2024, doi: 10.3390/math12111700.

[13] Z. Shao, M. N. Ahmad, and A. Javed, "Comparison of Random Forest and XGBoost Classifiers Using Integrated Optical and SAR Features for Mapping Urban Impervious Surface," *Remote Sens (Basel)*, vol. 16, no. 4, Feb. 2024, doi: 10.3390/rs16040665.

[14] R. Abedi, R. Costache, H. Shafizadeh-Moghadam, and Q. B. Pham, "Flash-flood susceptibility mapping based on XGBoost, random forest and boosted regression trees," *Geocarto Int*, vol. 37, no. 19, pp. 5479–5496, 2022, doi: 10.1080/10106049.2021.1920636.

[15] Y. Zhou, W. Shao, F. Nunziata, W. Wang, and C. Li, "An Algorithm to Retrieve Range Ocean Current Speed under Tropical Cyclone Conditions from Sentinel-1 Synthetic Aperture Radar Measurements Based on XGBoost," *Remote Sens (Basel)*, vol. 16, no. 17, Sep. 2024, doi: 10.3390/rs16173271.

[16] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4514–4523, Jul. 2022, doi: 10.1016/j.jksuci.2020.10.013.

[17] R. Wang, J. Zhang, B. Shan, M. He, and J. Xu, "XGBoost Machine Learning Algorithm for Prediction of Outcome in Aneurysmal Subarachnoid Hemorrhage," *Neuropsychiatr Dis Treat*, vol. 18, pp. 659–667, 2022, doi: 10.2147/NDT.S349956.

[18] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, Dec. 2021, doi: 10.3390/informatics8040079.

[19] T. Kee and W. K. O. Ho, "Optimizing Machine Learning Models for Urban Sciences: A Comparative Analysis of Hyperparameter Tuning Methods," *Urban Science*, vol. 9, no. 9, p. 348, Aug. 2025, doi: 10.3390/urbansci9090348.

[20] A. R. M. Rom, N. Jamil, and S. Ibrahim, "Multi objective hyperparameter tuning via random search on deep learning models," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 22, no. 4, pp. 956–968, Aug. 2024, doi: 10.12928/TELKOMNIKA.v22i4.25847.

[21] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 6, pp. 198–207, Dec. 2021, doi: 10.22266/ijies2021.1231.19.

[22] T. Kavzoglu and A. Teke, "Advanced hyperparameter optimization for improved spatial prediction of shallow landslides using extreme gradient boosting (XGBoost)," *Bulletin of Engineering Geology and the Environment*, vol. 81, no. 5, May 2022, doi: 10.1007/s10064-022-02708-w.

[23] Y. Kim, S. Steen, and H. Muri, "A novel method for estimating missing values in ship principal data," *Ocean Engineering*, vol. 251, May 2022, doi: 10.1016/j.oceaneng.2022.110979.

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

[24] C. Ribeiro and A. A. Freitas, “A data-driven missing value imputation approach for longitudinal datasets,” *Artif Intell Rev*, vol. 54, no. 8, pp. 6277–6307, Dec. 2021, doi: 10.1007/s10462-021-09963-5.

[25] D. S. Lee and S. Y. Son, “Weighted Average Ensemble-Based PV Forecasting in a Limited Environment with Missing Data of PV Power,” *Sustainability (Switzerland)* , vol. 16, no. 10, May 2024, doi: 10.3390/su16104069.

[26] M. Mera-Gaona, U. Neumann, R. Vargas-Canas, and D. M. López, “Evaluating the impact of multivariate imputation by MICE in feature selection,” *PLoS One*, vol. 16, no. 7 July, Jul. 2021, doi: 10.1371/journal.pone.0254720.

[27] J. Yu, R. Pan, and Y. Zhao, “High-Dimensional, Small-Sample Product Quality Prediction Method Based on MIC-Stacking Ensemble Learning,” *Applied Sciences (Switzerland)*, vol. 12, no. 1, Jan. 2022, doi: 10.3390/app12010023.

[28] D. Lee and K. Kim, “An efficient method to determine sample size in oversampling based on classification complexity for imbalanced data,” *Expert Syst Appl*, vol. 184, Dec. 2021, doi: 10.1016/j.eswa.2021.115442.

[29] A. S. Tarawneh, A. B. Hassanat, G. A. Altarawneh, and A. Almuhaimeed, “Stop Oversampling for Class Imbalance Learning: A Review,” *IEEE Access*, vol. 10, pp. 47643–47660, 2022, doi: 10.1109/ACCESS.2022.3169512.

[30] F. Duan, S. Zhang, Y. Yan, and Z. Cai, “An Oversampling Method of Unbalanced Data for Mechanical Fault Diagnosis Based on MeanRadius-SMOTE,” *Sensors*, vol. 22, no. 14, Jul. 2022, doi: 10.3390/s22145166.

[31] N. Anđelić and S. Baressi Šegota, “Achieving High Accuracy in Android Malware Detection through Genetic Programming Symbolic Classifier,” *Computers*, vol. 13, no. 8, Aug. 2024, doi: 10.3390/computers13080197.

[32] P. Gnip, L. Vokorokos, and P. Drotár, “Selective oversampling approach for strongly imbalanced data,” *PeerJ Comput Sci*, vol. 7, pp. 1–22, 2021, doi: 10.7717/PEERJ-CS.604.

[33] T. K. Dang, T. C. Tran, L. M. Tuan, and M. V. Tiep, “Machine learning based on resampling approaches and deep reinforcement learning for credit card fraud detection systems,” *Applied Sciences (Switzerland)*, vol. 11, no. 21, Nov. 2021, doi: 10.3390/app112110004.

[34] C. D. Nguyen, J. B. Carlin, and K. J. Lee, “Practical strategies for handling breakdown of multiple imputation procedures,” *Emerg Themes Epidemiol*, vol. 18, no. 1, Dec. 2021, doi: 10.1186/s12982-021-00095-3.

[35] H. El Azhari *et al.*, “Predicting the Production and Depletion of Rare Earth Elements and Their Influence on Energy Sector Sustainability through the Utilization of Multilevel Linear Prediction Mixed-Effects Models with R Software,” *Sustainability*, vol. 16, no. 5, p. 1951, Feb. 2024, doi: 10.3390/su16051951.

[36] H. S. Laqueur, A. B. Shev, and R. M. C. Kagawa, “SuperMICE: An Ensemble Machine Learning Approach to Multiple Imputation by Chained Equations,” *Am J Epidemiol*, vol. 191, no. 3, pp. 516–525, Mar. 2022, doi: 10.1093/aje/kwab271.

[37] L. J. Beesley and J. M. G. Taylor, “A stacked approach for chained equations multiple imputation incorporating the substantive model,” *Biometrics*, vol. 77, no. 4, pp. 1342–1354, Dec. 2021, doi: 10.1111/biom.13372.

[38] J. K. Essel, J. A. Mensah, E. Ocran, and L. Asiedu, “On the search for efficient face recognition algorithm subject to multiple environmental constraints,” *Heliyon*, vol. 10, no. 7, Apr. 2024, doi: 10.1016/j.heliyon.2024.e28568.

[39] N. U. Okafor and D. T. Delaney, “Missing Data Imputation on IoT Sensor Networks: Implications for on-Site Sensor Calibration,” *IEEE Sens J*, vol. 21, no. 20, pp. 22833–22845, Oct. 2021, doi: 10.1109/JSEN.2021.3105442.

[40] F. B. Hamzah, F. Mohamad Hamzah, S. F. Mohd Razali, and A. El-Shafie, “Multiple imputations by chained equations for recovering missing daily streamflow observations: a case study of Langat River basin in Malaysia,” *Hydrological Sciences Journal*, vol. 67, no. 1, pp. 137–149, 2022, doi: 10.1080/02626667.2021.2001471.

[41] F. B. Hamzah, F. M. Hamzah, S. F. M. Razali, and H. Samad, “A comparison of multiple imputation methods for recovering missing data in hydrological studies,” *Civil Engineering Journal (Iran)*, vol. 7, no. 9, pp. 1608–1619, Sep. 2021, doi: 10.28991/cej-2021-03091747.

[42] L. Song and G. Guo, “Full Information Multiple Imputation for Linear Regression Model with Missing Response Variable,” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 1, pp. 77–81, 2024.

[43] R. K. Kim *et al.*, “Data integration of National Dose Registry and survey data using multivariate imputation by chained equations,” *PLoS One*, vol. 17, no. 6 June, Jun. 2022, doi: 10.1371/journal.pone.0261534.

[44] N. A. M. Pauzi, Y. B. Wah, S. M. Deni, S. K. N. A. Rahim, and Suhartono, “Comparison of single and mice imputation methods for missing values: A simulation study,” *Pertanika J Sci Technol*, vol. 29, no. 2, pp. 979–998, 2021, doi: 10.47836/pjst.29.2.15.

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

[45] G. Liu, T. Zhang, H. Dai, X. Cheng, and D. Yang, “ResInceptNet-SA: A Network Traffic Intrusion Detection Model Fusing Feature Selection and Balanced Datasets,” *Applied Sciences (Switzerland)*, vol. 15, no. 2, Jan. 2025, doi: 10.3390/app15020956.

[46] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, “Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study,” *Applied Sciences*, vol. 15, no. 4, p. 1903, Feb. 2025, doi: 10.3390/app15041903.

[47] J. Fonseca, G. Douzas, and F. Bacao, “Improving imbalanced land cover classification with k-means smote: Detecting and oversampling distinctive minority spectral signatures,” *Information (Switzerland)*, vol. 12, no. 7, Jul. 2021, doi: 10.3390/info12070266.

[48] T. Wongvorachan, S. He, and O. Bulut, “A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining,” *Information (Switzerland)*, vol. 14, no. 1, Jan. 2023, doi: 10.3390/info14010054.

[49] I. Ul Hassan, R. H. Ali, Z. Ul Abideen, T. A. Khan, and R. Kouatly, “Significance of Machine Learning for Detection of Malicious Websites on an Unbalanced Dataset,” *Digital*, vol. 2, no. 4, pp. 501–519, Dec. 2022, doi: 10.3390/digital2040027.

[50] M. Alrumaidhi, M. M. G. Farag, and H. A. Rakha, “Comparative Analysis of Parametric and Non-Parametric Data-Driven Models to Predict Road Crash Severity among Elderly Drivers Using Synthetic Resampling Techniques,” *Sustainability (Switzerland)*, vol. 15, no. 13, Jul. 2023, doi: 10.3390/su15139878.

[51] C. Azad, B. Bhushan, R. Sharma, A. Shankar, K. K. Singh, and A. Khamparia, “Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus,” in *Multimedia Systems*, Springer Science and Business Media Deutschland GmbH, Aug. 2022, pp. 1289–1307. doi: 10.1007/s00530-021-00817-2.

[52] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, “A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data,” *Inf Sci (N Y)*, vol. 572, pp. 574–589, Sep. 2021, doi: 10.1016/j.ins.2021.02.056.

[53] J. Li, Q. Zhu, Q. Wu, and Z. Fan, “A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors,” *Inf Sci (N Y)*, vol. 565, pp. 438–455, Jul. 2021, doi: 10.1016/j.ins.2021.03.041.

[54] S. P. Kenaka, A. Cakravastia, A. Ma’ruf, and R. T. Cahyono, “Enhancing Intermittent Spare Part Demand Forecasting: A Novel Ensemble Approach with Focal Loss and SMOTE,” *Logistics*, vol. 9, no. 1, p. 25, Feb. 2025, doi: 10.3390/logistics9010025.

[55] J. Liu, F. Tian, A. Zhao, W. Zheng, and W. Cao, “Logging Lithology Discrimination with Enhanced Sampling Methods for Imbalance Sample Conditions,” *Applied Sciences (Switzerland)*, vol. 14, no. 15, Aug. 2024, doi: 10.3390/app14156534.

[56] S. Wang, Y. Dai, J. Shen, and J. Xuan, “Research on expansion and classification of imbalanced data based on SMOTE algorithm,” *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-03430-5.

[57] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, “A Deep Learning Model for Network Intrusion Detection with Imbalanced Data,” *Electronics (Switzerland)*, vol. 11, no. 6, Mar. 2022, doi: 10.3390/electronics11060898.

[58] S. K. Ganesan, P. Velusamy, S. Rajendran, R. Sakthivel, M. Bose, and B. S. Inbaraj, “ZooCNN: A Zero-Order Optimized Convolutional Neural Network for Pneumonia Classification Using Chest Radiographs,” *J Imaging*, vol. 11, no. 1, Jan. 2025, doi: 10.3390/jimaging11010022.

[59] M. R. Askari, M. Abdel-Latif, M. Rashid, M. Sevil, and A. Cinar, “Detection and Classification of Unannounced Physical Activities and Acute Psychological Stress Events for Interventions in Diabetes Treatment,” *Algorithms*, vol. 15, no. 10, Oct. 2022, doi: 10.3390/a15100352.

[60] M. Glučina, A. Lorencin, N. Anđelić, and I. Lorencin, “Cervical Cancer Diagnostics Using Machine Learning Algorithms and Class Balancing Techniques,” *Applied Sciences (Switzerland)*, vol. 13, no. 2, Jan. 2023, doi: 10.3390/app13021061.

[61] R. Wang, F. Liu, and Y. Bai, “A Software Defect Prediction Method That Simultaneously Addresses Class Overlap and Noise Issues after Oversampling,” *Electronics (Switzerland)*, vol. 13, no. 20, Oct. 2024, doi: 10.3390/electronics13203976.

[62] R. M. Saputra *et al.*, “Improving Cervical Cancer Classification Using ADASYN and Random Forest with GridSearchCV Optimization,” vol. 16, no. 01, 2025, doi: 10.35970/infotekmesin.v16i1.2552.

[63] D. Chen, W. Li, and J. Fang, “Blending-Based Ensemble Learning Low-Voltage Station Area Theft Detection,” *Energies (Basel)*, vol. 18, no. 1, Jan. 2025, doi: 10.3390/en18010031.

[64] A. Akinjole, O. Shobayo, J. Popoola, O. Okoyeigbo, and B. Ogunleye, “Ensemble-Based Machine Learning Algorithm for Loan Default

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Risk Prediction," *Mathematics*, vol. 12, no. 21, Nov. 2024, doi: 10.3390/math12213423.

[65] S. Chalichalamala, N. Govindan, and R. Kasarapu, "Logistic Regression Ensemble Classifier for Intrusion Detection System in Internet of Things," *Sensors*, vol. 23, no. 23, Dec. 2023, doi: 10.3390/s23239583.

[66] V. R. Joseph, "Optimal ratio for data splitting," *Stat Anal Data Min*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.

[67] Q. H. Nguyen *et al.*, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/4832864.

[68] Y. Aydın, C. Cakiroglu, G. Bekdaş, and Z. W. Geem, "Explainable Ensemble Learning and Multilayer Perceptron Modeling for Compressive Strength Prediction of Ultra-High-Performance Concrete," *Biomimetics*, vol. 9, no. 9, p. 544, Sep. 2024, doi: 10.3390/biomimetics9090544.

[69] X. Zhu, J. Chu, K. Wang, S. Wu, W. Yan, and K. Chiam, "Prediction of rockhead using a hybrid N-XGBoost machine learning framework," *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 13, no. 6, pp. 1231–1245, Dec. 2021, doi: 10.1016/j.jrmge.2021.06.012.

[70] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3360–3379, 2023, doi: 10.1080/10494820.2021.1928235.

[71] M. Ma *et al.*, "XGBoost-based method for flash flood risk assessment," *J Hydrol (Amst)*, vol. 598, Jul. 2021, doi: 10.1016/j.jhydrol.2021.126382.

[72] S. Thongsuwan, S. Jaiyen, A. Padcharoen, and P. Agarwal, "ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost," *Nuclear Engineering and Technology*, vol. 53, no. 2, pp. 522–531, Feb. 2021, doi: 10.1016/j.net.2020.04.008.

[73] G. Qi and B. Liu, "Production Feature Analysis of Global Onshore Carbonate Oil Reservoirs Based on XGBoost Classier," *Processes*, vol. 12, no. 6, Jun. 2024, doi: 10.3390/pr12061137.

[74] Y. Wang *et al.*, "Short-term load forecasting of industrial customers based on SVMD and XGBoost," *International Journal of Electrical Power and Energy Systems*, vol. 129, Jul. 2021, doi: 10.1016/j.ijepes.2021.106830.

[75] N. H. Tiep *et al.*, "A New Hyperparameter Tuning Framework for Regression Tasks in Deep Neural Network: Combined-Sampling Algorithm to Search the Optimized Hyperparameters," *Mathematics*, vol. 12, no. 24, Dec. 2024, doi: 10.3390/math12243892.

[76] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," Mar. 01, 2023, *John Wiley and Sons Inc*. doi: 10.1002/widm.1484.

[77] R. Hossain and D. Timmer, "Machine Learning Model Optimization with Hyper Parameter Tuning Approach," 2021.

[78] R. Valarmathi and T. Sheela, "Heart disease prediction using hyper parameter optimization (HPO) tuning," *Biomed Signal Process Control*, vol. 70, Sep. 2021, doi: 10.1016/j.bspc.2021.103033.

[79] X. Zhong *et al.*, "Automatic Classification of All-Sky Nighttime Cloud Images Based on Machine Learning," *Electronics (Switzerland)*, vol. 13, no. 8, Apr. 2024, doi: 10.3390/electronics13081503.

[80] M. K. Suryadi, R. Herteno, S. W. Saputro, M. R. Faisal, and R. A. Nugroho, "Comparative Study of Various Hyperparameter Tuning on Random Forest Classification With SMOTE and Feature Selection Using Genetic Algorithm in Software Defect Prediction," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 137–147, Mar. 2024, doi: 10.35882/jeeemi.v6i2.375.

[81] H. Ghinaya, R. Herteno, M. R. Faisal, A. Farmadi, and F. Indriani, "Analysis of Important Features in Software Defect Prediction Using Synthetic Minority Oversampling Techniques (SMOTE), Recursive Feature Elimination (RFE) and Random Forest," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 3, pp. 276–288, May 2024, doi: 10.35882/jeeemi.v6i3.453.

[82] M. M. Muraru, Z. Simó, and L. B. Iantovics, "Cervical Cancer Prediction Based on Imbalanced Data Using Machine Learning Algorithms with a Variety of Sampling Methods," *Applied Sciences (Switzerland)*, vol. 14, no. 22, Nov. 2024, doi: 10.3390/app142210085.

[83] M. C. Hinojosa Lee, J. Braet, and J. Springael, "Performance Metrics for Multilabel Emotion Classification: Comparing Micro, Macro, and Weighted F1-Scores," *Applied Sciences (Switzerland)*, vol. 14, no. 21, Nov. 2024, doi: 10.3390/app14219863.

[84] Ž. Vujović, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[85] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies (Basel)*,

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

vol. 9, no. 4, Dec. 2021, doi: 10.3390/technologies9040081.

[86] N. Aida, T. H. Saragih, D. Kartini, R. A. Nugroho, and D. T. Nugrahadi, “Comparison of Extreme Machine Learning and Hidden Markov Model Algorithm in Predicting The Recurrence Of Differentiated Thyroid Cancer Using SMOTE,” *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 4, pp. 429–444, Oct. 2024, doi: 10.35882/jeeemi.v6i4.467.

[87] S. Farhadpour, T. A. Warner, and A. E. Maxwell, “Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices,” *Remote Sens (Basel)*, vol. 16, no. 3, Feb. 2024, doi: 10.3390/rs16030533.

[88] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, “Confidence interval for micro-averaged F 1 and macro-averaged F 1 scores,” *Applied Intelligence*, vol. 52, no. 5, pp. 4961–4972, Mar. 2022, doi: 10.1007/s10489-021-02635-5.

[89] N. Namdev and N. Tomar, “Ad Click Prediction: A Comparative Evaluation of Logistic Regression and Performance Metrics,” *Int J Res Appl Sci Eng Technol*, vol. 11, no. 7, pp. 1514–1523, Jul. 2023, doi: 10.22214/ijraset.2023.54914.

[90] F. S. Nahm, “Receiver operating characteristic curve: overview and practical use for clinicians,” *Korean J Anesthesiol*, vol. 75, no. 1, pp. 25–36, Feb. 2022, doi: 10.4097/kja.21209.

[91] U. K. Lilhore *et al.*, “Hybrid Model for Detection of Cervical Cancer Using Causal Analysis and Machine Learning Techniques,” *Comput Math Methods Med*, vol. 2022, 2022, doi: 10.1155/2022/4688327.

[92] H. Karamti *et al.*, “Improving Prediction of Cervical Cancer Using KNN Imputed SMOTE Features and Multi-Model Ensemble Learning Approach,” *Cancers (Basel).*, vol. 15, no. 17, Sep. 2023, doi: 10.3390/cancers15174412.

Author Biography

Mita Azzahra Ramadhan is currently a Computer Science student at the Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University, having commenced her studies in 2020. She has developed a strong passion for data science and machine learning, with a particular focus on applications in medical data classification. Throughout her academic journey, she has engaged in several projects involving advanced techniques for handling imbalanced datasets and improving prediction accuracy. Her final undergraduate project involved a comparative analysis of two popular oversampling methods, SMOTE and ADASYN, for cervical cancer classification using the XGBoost algorithm optimized with Random Search hyperparameter tuning and data imputation via the MICE method.

Traindo Hamonangan Saragih is currently a lecturer at the Department of Computer Science, Lambung Mangkurat University. He is deeply involved in the academic world, with a main focus on various aspects of Data Science. His academic journey began with completing his undergraduate studies in Informatics at Brawijaya University, Malang, which he successfully completed in 2016. To deepen his understanding in Computer Science, he continued his education at the master's level at the same university and earned a master's degree in 2018. His research interests center on Data Science, where he continues to develop her expertise in data analysis, predictive modeling, and the application of data-driven technologies. As an academic, he is active in scientific research and collaboration to develop innovative solutions applicable across various fields. He also seeks to contribute to the development of science through publications and academic discussions. Email: triando.saragih@ulm.ac.id Orcid ID: 0000-0003-4346 3323.

Dwi Kartini is a lecturer with a strong basis in computer science. Having obtained both her bachelor's and master's degrees from the Faculty of Computer Science at Putra Indonesia "YPTK" Padang, Indonesia, she has dedicated her career to teaching and research in this field. As a lecturer in the Department of Computer Science, Homepage: jeeemi.org, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University in Banjarbaru, Indonesia, she imparts her knowledge to students through courses such as linear algebra, discrete mathematics, and research methodologies. Her scholarly interests converge on the practical applications of Artificial Intelligence and Data Mining, making her a valuable asset to the academic community.

Muhammad Itqan Mazdadi is a lecturer in the Department of Computer Science at University, Lambung Mangkurat, with research interests primarily in Data Science and Computer Networking. His academic journey began at Lambung Mangkurat

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

University, where he earned his undergraduate degree in Computer Science in 2013. To further deepen his expertise, he pursued and successfully completed a master’s degree in Informatics from the Islamic University of Indonesia, Yogyakarta. Currently, he serves as Secretary in the Computer Science Department at Lambung Mangkurat University, contributing to both academic and administrative responsibilities. His research focuses on exploring advancements in data-driven technologies and network systems. Through his role as a lecturer and researcher, he actively engages in academic development, fostering innovation in his field. His dedication to education and research enables him to support students and collaborate on projects that enhance technological progress. Email: mazdadi@ulm.ac.id Orcid ID: 0000-0002-8710-4616.

Muliadi is a lecturer in the Department of Computer Science at Lambung Mangkurat University, specializing in Artificial Intelligence, Decision Support Systems, and Data Science. His academic journey commenced with earning a bachelor's degree in Informatics Engineering from STMIK Akakomin in 2004. To further enhance his knowledge, he pursued and successfully obtained a master's degree in Computer Science from Gadjah Mada University in 2009. With a strong foundation in Data Science, he possesses extensive expertise in analyzing and interpreting complex datasets. Additionally, he has valuable skills in Start-up Business Development, Digital Entrepreneurship, and Data Management. His experience allows him to contribute significantly to both academic and practical applications of technology. Beyond his role as a lecturer, he actively engages in research and collaborative projects to advance technological innovation. Through his expertise, he strives to bridge the gap between academia and industry, fostering solutions that drive digital transformation and business growth. Email: muliadi@ulm.ac.id Orcid ID: 0000-0003-2871-9482.

Manuscript received 13 November 2025; Revised 10 January 2026; Accepted 22 January 2026; Available online 24 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1415

Copyright © 2026 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).