



Research Article

An AI-Driven Framework for Diabetes Prediction and Classification Using Medical Data, SMOTE, and Grid Search CV Optimization

Devashish Chauhan¹ , Rakshit Sharma² , Shyam Ji³  and Suraj Singh Patwal^{4*} 

^{1&2}Department of Computer Science and Engineering, Graphic Era, Dehradun, India

³Department of Electronics and Communication, NIT Hamirpur, India

⁴Department of Mechanical Engineering, IIT Patna, India

Article Information

Abstract

Article History

Received: 24 May 2026

Revised: 4 July 2026

Accepted: 24 August 2026

Published online: 7 September 2026

Keywords

Supervised Machine Learning

Classification Task

Algorithms

SMOTE

Grid Search CV

Diabetes Prediction

Correspondence*

patwalsuraj11@gmail.com

ORCID

Devashish Chauhan 

<https://orcid.org/0009-0002-8868-6213>

Rakshit Sharma 

<https://orcid.org/0009-0006-9990-6145>

Shyam Ji 

<https://orcid.org/0009-0007-9282-9711>

Suraj Singh Patwal 

<https://orcid.org/0009-0006-9013-6885>

In India, approximately 101 million people are diabetic, while around 136 million are prediabetic, making early diagnosis essential for preventing severe health complications. The ML field of AI can emerge as an alternative for diagnosing diabetes. In this study, various algorithms are employed for diabetes prediction using the Pima Indian Diabetes Dataset. Hyperparameter optimization was performed using Grid Search Cross Validation (Grid Search CV) to identify the optimal model parameters. Since the dataset is imbalanced, experiments were conducted on both the original imbalanced dataset and a balanced synthesized dataset generated using SMOTE, which significantly reduced False Negative (FN) predictions across most algorithms. Furthermore, Random Forest and Naive Bayes achieved the highest recall of 72.73% after applying SMOTE, demonstrating improved effectiveness in identifying diabetic patients.

© 2026 Centre for Research and Innovation (CRI). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

I. INTRODUCTION

Diabetes mellitus occurs because the pancreas is not able to release insulin to digest sugar and restrict the supply of glucose to the cells. According to global health reports, the number of diabetic patients is continuously rising due to unhealthy lifestyles, obesity, lack of physical activity, stress, aging populations, and genetic factors. Early detection and accurate diagnosis are therefore essential to reduce complications and improve patient outcomes. Traditional methods for diagnosing diabetes involve laboratory tests. Although these diagnostic techniques are clinically effective, they may require repeated hospital visits, skilled medical professionals, and considerable time for analysis. In many

cases, manual interpretation of medical data can be complex, prone to human error, and time-consuming when a large number of tests are required. As a result, machine learning can be used to diagnose diabetes. Using machine learning in diabetes prediction offers several advantages over traditional diagnostic approaches. First, models can process a huge number of diagnoses of the given diseases faster and more accurately than humans. Second, these models can identify complex relationships between multiple clinical attributes that may not be easily recognized through manual analysis. Early prediction enables patients to adopt preventive measures such as lifestyle modifications, proper medication, regular exercise, and dietary control before severe complications occur.

II. LITERATURE REVIEW

Artificial intelligence is used to diagnose diabetes, in which various machine learning models predict whether a person is diabetic based on various independent parameters. Kumar *et al.*, [1] use a Bayesian model and a multilayer perceptron in an ensemble model to predict whether a person has diabetes. They also perform feature engineering and select the features that deliver maximum accuracy in the model. In many cases, researchers use the same dataset, named the Pima Indian Diabetes Dataset.

Mujumdar *et al.*, [2] create their own dataset with 800 data points and 10 features, apply classification algorithms to predict diabetes, and compare the results with the Pima Indian Diabetes Dataset. They achieve an accuracy of 98.8% with the AdaBoost classifier using a pipeline, while with algorithms only they reach up to 96% accuracy using logistic regression on their own dataset. In the imbalanced Pima Indian Dataset, they achieve up to 77% accuracy; it requires balancing using SMOTE before applying the algorithms, as done in this research paper. Ali *et al.*, [3] provide an overview related to this topic, cover the algorithms that can be used to perform this task, present a flow chart from data collection to deployment, and suggest various techniques for collecting the dataset.

Khokhar *et al.*, [4] use a deep learning technique with SMOTE to tackle the class imbalance problem. The dataset they use is the Diabetes Binary Health Indicators Dataset, which is different from the Pima Indian Diabetes Dataset, and their model gives an accuracy of 92.5%. Similarly, Mondal *et al.*, [5] apply all the algorithms, and the Naive Bayes algorithm gives the highest accuracy on the Pima dataset. Tigga *et al.*, [6] created their own dataset of 952 patients and performed classification using six algorithms, with the most accurate model achieving an accuracy of 94%.

Hasan *et al.*, [7] use a combination of k-fold cross-validation and Grid Search CV techniques to train the algorithms and use a deep learning algorithm to predict diabetes. Febrian *et al.*, [8] use Naive Bayes and KNN to perform experiments and determine accuracy and precision after 10% training is complete using the Pima Indian Diabetes Dataset. Yacoub *et al.*, [9], Sujon *et al.*, [10], and Helmud *et al.*, [11] describe various evaluation metrics.

Kotsiantis *et al.*, [12] and Alzubi *et al.*, [13] describe a classification task using various machine learning algorithms. Longadge *et al.*, [14] provide an overview of the class imbalance classification task, and Devi *et al.*, [15] use an undersampling technique as a solution to the class imbalance problem. Ahmad *et al.*, [16] perform the diagnosis of heart diseases using machine learning algorithms with or without the Grid Search CV technique. Aprilliandhika *et al.*, [17] perform Grid Search CV in stroke prediction using KNN and SVM machine learning algorithms. Pradipta *et al.*, [18] provide a review of SMOTE.

III. METHODOLOGY

In this research, the dataset is taken from Kaggle. It contains 768 data points, eight independent columns or features, and one dependent output column, which has classes indicating diabetes or not. The eight independent features and one dependent feature are shown in Figure 2. The output class is imbalanced, with fewer patients having diabetes than normal people, and the dataset is already clean, so no data preprocessing is required.

TABLE I THE TABLE SHOWS AN IMBALANCE OF CLASSES IN THE PIMA DIABETES DATASET

Classes	Count of class
Diabetic person	268
Normal person	500

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Fig. 1 Data Frame of the Pima Dataset

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Pregnancies                          768 non-null    int64
1   Glucose                             768 non-null    int64
2   BloodPressure                       768 non-null    int64
3   SkinThickness                      768 non-null    int64
4   Insulin                            768 non-null    int64
5   BMI                                768 non-null    float64
6   DiabetesPedigreeFunction            768 non-null    float64
7   Age                                768 non-null    int64
8   Outcome                            768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB

```

Fig. 2 Dataset is Clean and has no Null Values

The heat map shows the high correlation of BMI, Age, Glucose, Insulin, and Diabetes Pedigree Function with the output column.

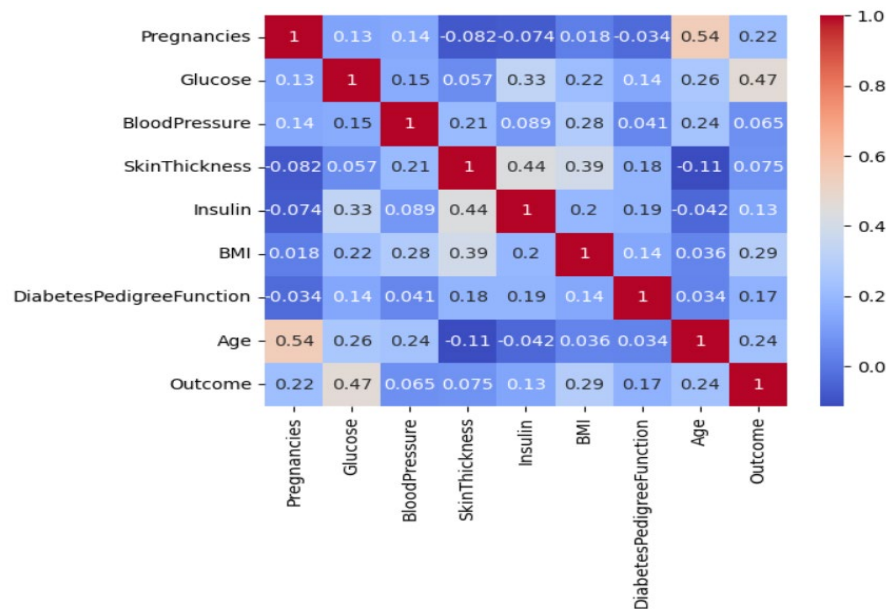


Fig. 3 Correlation of Each Column with the Other Columns

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479
std	3.369578	31.972618	19.355807	15.952218	115.244002
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000
75%	6.000000	140.250000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%	36.600000	0.626250	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

Fig. 4 Description of Each Column in the Dataset

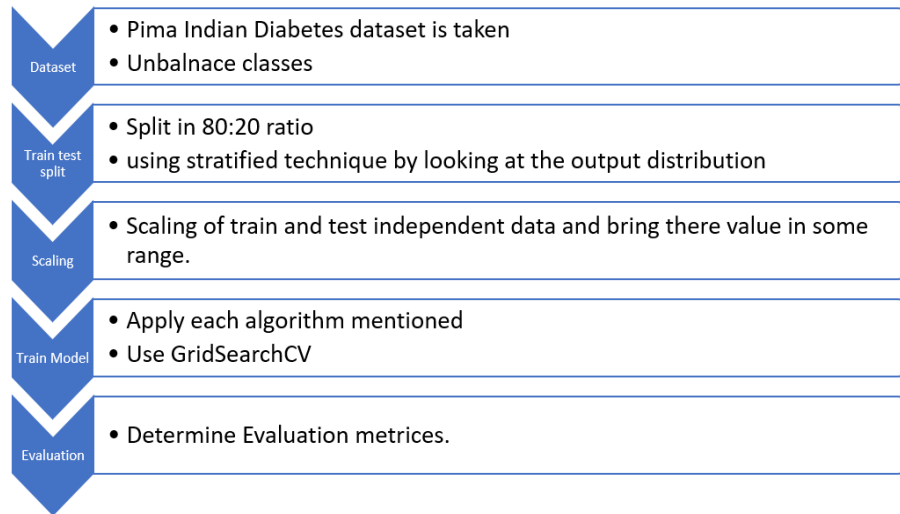


Fig. 5 Flow Diagram of the First Approach Without Balancing the Classes

Our first approach consists of splitting the dataset in the ratio 80:20 by stratifying on the output column, which maintains the same train and test distribution in terms of output class distribution. Various classification-task machine learning

algorithms are trained on the training dataset while keeping the training dataset imbalanced. The algorithms use Grid Search CV, which takes a range of parameters and predicts the best possible model.

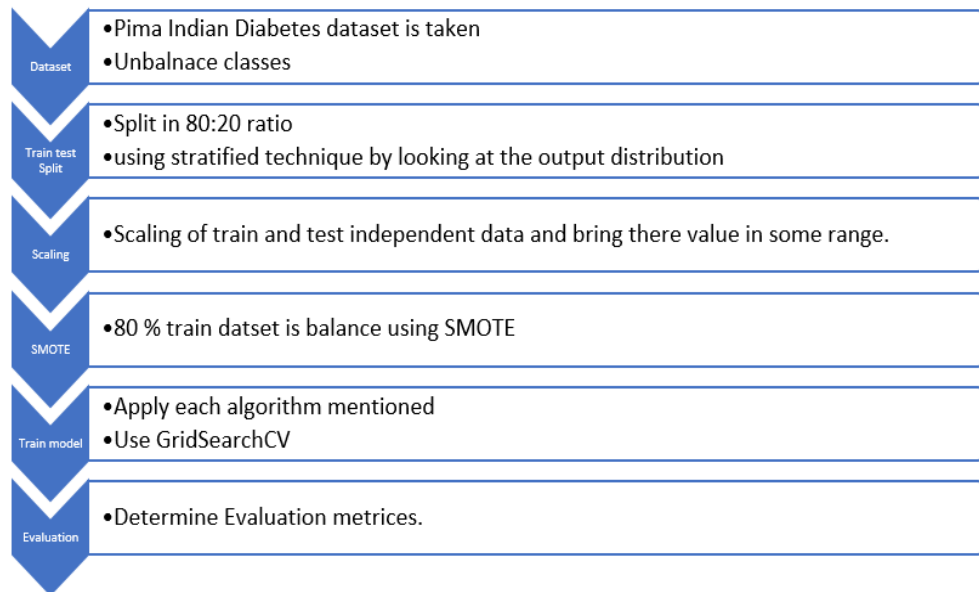


Fig. 6 Flow Diagram of the Second Approach with Balanced Classes

After the above approach, the use of SMOTE (Synthetic Minority Over-sampling Technique) synthesizes the training dataset and balances it by increasing the samples of the minority class. We then apply the above-mentioned algorithms with Grid Search CV. From this approach, whichever model performs well is used to train on various combinations of the features using feature engineering and predict its accuracy by inserting various combinations of the independent features.

Grid Search CV: It is a hyperparameter tuning technique that takes a parameter grid, tries every possible combination of parameters, performs cross-validation of the training dataset,

predicts the best possible parameter combination and score, and uses it with every machine learning algorithm mentioned above.

SMOTE: It is a technique in which an imbalanced dataset is balanced by synthesizing the minority classes and increasing their samples using the steps below.

SMOTE's Steps: It chooses k-nearest-neighbor data points from the minority class and, using the k data points, performs interpolation using these points. It continues creating new sample points in the minority class until both classes have the same distribution in the dataset.

Evaluation Metrics are used to evaluate our model. The following metrics are used in this paper:

Accuracy: It is an evaluation metric that is the ratio of the sum of true positives and true negatives to the sum of all possible outcomes of the output.

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

Precision: It is defined as the ratio of true positives to the sum of true positives and false positives and is used for detecting fraud and spam.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall: It is defined as the ratio of true positives to the sum of true positives and false negatives and can be used for the diagnosis of diseases.

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1 Score: It is defined as the hyperbolic mean of recall and precision.

$$\text{F1 score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

Confusion matrix: It is a representation of all the types of outputs from the test dataset when the algorithm is applied, as shown below.

It is a confusion matrix corresponding to the scikit-learn library.

[[TP FP]
[FN TN]]

IV. RESULTS AND OBSERVATIONS

Here, we split the dataset in an 80:20 ratio, corresponding to training and testing, and Table I shows the distribution of classes for the whole dataset. In the evaluation task, the distribution of the test dataset is important.

TABLE II DISTRIBUTION OF CLASSES IN THE TEST DATASET

Classes	Counts
Diabetes	54
Normal	100

In the observation, we train the various algorithms mentioned in the methodology section using Grid Search CV optimization with and without the SMOTE technique. Our dataset is well distributed but imbalanced, and the evaluation parameters are presented. Apart from the best model parameters, the confusion matrices are also displayed.

TABLE IV BEST PARAMETERS CORRESPONDING TO EACH ALGORITHM WITH OR WITHOUT APPLYING THE SMOTE TECHNIQUE IN GRID SEARCH CV OPTIMIZATION

Algorithms	Without SMOTE	With SMOTE
Logistic regression	{'C': 1, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'}	{'C': 0.1, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'}
Decision tree	{'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10}	{'criterion': 'entropy', 'max_depth': 15, 'min_samples_leaf': 4, 'min_samples_split': 2}
Random forest	{'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}	{'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Support vector machine	{'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}	{'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}
Naive Bayes	{'var_smoothing': 1e-09}	{'var_smoothing': 1e-09}
K-nearest neighbor	{'metric': 'euclidean', 'n_neighbors': 9, 'weights': 'uniform'}	{'metric': 'euclidean', 'n_neighbors': 5, 'weights': 'distance'}

TABLE V CONFUSION MATRICES CORRESPONDING TO ALL ALGORITHMS WITH OR WITHOUT APPLYING THE SMOTE TECHNIQUE

Algorithms	Without SMOTE	With SMOTE
Logistic regression	[[82 18] [26 28]]	[[70 29] [15 40]]
Decision tree	[[83 17] [15 39]]	[[78 21] [17 38]]
Random forest	[[86 14] [23 31]]	[[74 25] [15 40]]
Support vector machine	[[83 17] [26 28]]	[[68 31] [23 32]]
Naive Bayes	[[74 26] [19 35]]	[[71 28] [15 40]]
K-nearest neighbor	[[83 17] [24 30]]	[[66 33] [17 38]]

TABLE VI VARIOUS EVALUATION METRICS WITHOUT SMOTE

Algorithms	Accuracy	Precision	Recall	F1-score
Logistic regression	71.43%	60.87%	51.85%	56%
Decision tree	79.22%	69.64%	72.22%	70.91%
Random forest	75.97%	68.89%	57.41%	62.63%
Support vector machine	72.08%	62.22%	51.85%	56.56%
Naive Bayes	70.78%	57.38%	64.81%	60.87%
K-nearest neighbor	73.38%	63.83%	55.55%	59.40%

TABLE VII VARIOUS EVALUATION METRICS WITH SMOTE

Algorithms	Accuracy	Precision	Recall	F1-score
Logistic regression	71.43%	57.97%	72.72%	64.52%
Decision tree	75.32%	64.41%	69.09%	66.67%
Random forest	74.02%	61.54%	72.73%	66.67%
Support vector machine	64.93%	50.79%	58.18%	54.24%
Naive Bayes	72.08%	58.82%	72.73%	65.04%
K-nearest neighbor	67.53%	53.52%	69.09%	60.32%

V. CONCLUSION

In diabetes prediction using machine learning, recall and false negative (FN) values are considered highly important evaluation factors because a false negative indicates that a diabetic patient is incorrectly predicted as non-diabetic. Such incorrect predictions may delay medical treatment and increase the risk of severe health complications. Therefore, minimizing FN values and maximizing recall are essential objectives in disease prediction systems. The results presented in Table 5 demonstrate that the use of the SMOTE technique significantly improved the performance of most machine learning algorithms. After applying SMOTE, the FN values were reduced in almost all algorithms compared with models trained without SMOTE. This indicates that balancing the dataset helped the models better identify diabetic patients. Furthermore, recall values increased for most algorithms when SMOTE was applied. Among all the models, Random Forest and Naive Bayes achieved the highest recall values using the SMOTE technique, showing their strong capability to correctly detect diabetic cases. Overall, the experimental results confirm that the integration of SMOTE with machine learning algorithms enhances diabetes prediction performance and improves the reliability of medical diagnosis systems.

Declaration of Conflicting Interests

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Use of Artificial Intelligence (AI)-Assisted Technology for Manuscript Preparation

The authors confirm that no AI-assisted technologies were used in the preparation or writing of the manuscript, and no images were altered using AI.

REFERENCES

- [1] A. Kumar Dewangan and P. Agrawal, "Classification of diabetes mellitus using machine learning techniques," *International Journal of Engineering and Applied Sciences*, vol. 2, no. 5, p. 257905, 2015.
- [2] A. Mujumdar and V. Vaidehi, "Diabetes prediction using machine learning algorithms," *Procedia Computer Science*, vol. 165, pp. 292–299, 2019.
- [3] C. S. M. Ali and O. S. Kareem, "Diabetes prediction using machine learning," *Asian Journal of Research in Computer Science*, vol. 18, no. 6, pp. 89–109, 2025.
- [4] P. B. Khokhar, V. Pentangelo, F. Palomba, and C. Gravino, "Towards transparent and accurate diabetes prediction using machine learning and explainable artificial intelligence," *arXiv preprint arXiv:2501.18071*, 2025.
- [5] S. Mondal and J. P. Choudhury, "Prediction of diabetes using machine learning models," *Science and Culture*, 2024.
- [6] N. P. Tigga and S. Garg, "Prediction of type 2 diabetes using machine learning classification methods," *Procedia Computer Science*, vol. 167, pp. 706–716, 2020.
- [7] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020.
- [8] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, "Diabetes prediction using supervised machine learning," *Procedia Computer Science*, vol. 216, pp. 21–30, 2023.
- [9] R. Yacoubi and D. Axman, "Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models," in *Proc. 1st Workshop on Evaluation and Comparison of NLP Systems*, 2020, pp. 79–91.
- [10] K. M. Sujon, R. Hassan, K. Choi, and M. A. Samad, "Accuracy, precision, recall, F1-score, or MCC? Empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models," *Journal of Big Data*, vol. 12, no. 1, p. 268, 2025.
- [11] E. Helmud, F. Fitriyani, and P. Romadiana, "Classification comparison performance of supervised machine learning random forest and decision tree algorithms using confusion matrix," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 13, no. 1, pp. 92–97, 2024.
- [12] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.
- [13] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview," *Journal of Physics: Conference Series*, vol. 1142, no. 1, Art. no. 012012, 2018.
- [14] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," *arXiv preprint arXiv:1305.1707*, 2013.
- [15] D. Devi, S. K. Biswas, and B. Purkayastha, "A review on solution to class imbalance problem: Undersampling approaches," in *Proc. 2020 Int. Conf. Computational Performance Evaluation (ComPE)*, 2020, pp. 626–631.
- [16] G. N. Ahmad, H. Fatima, S. Ullah, and A. S. Saidi, "Efficient medical diagnosis of human heart diseases using machine learning techniques with and without GridSearchCV," *IEEE Access*, vol. 10, pp. 80151–80173, 2022.
- [17] W. Aprilliandhika and F. F. Abdulloh, "Comparison of K-nearest neighbor and support vector machine algorithm optimization with grid search CV on stroke prediction," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 4, pp. 991–1000, 2024.
- [18] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, "SMOTE for handling imbalanced data problem: A review," in *Proc. 2021 6th Int. Conf. Informatics and Computing (ICIC)*, 2021, pp. 1–8.