IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

GENERATING SYNTHETIC PATIENT RECORDS WITH CTGAN TO IMPROVE CARDIOVASCULAR RISK PREDICTION

M. Manoj Kumar[1], Mrs. M. Santhikala[2], Dr. M. Kaliappan[3], Dr. E. Mariappan[4]

Department of Artificial Intelligence and Data Science,

Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India[1-4]

Abstract: Heart disease is among the top causes of death around the world, and catching it at an early stage can make a real difference in how patients are treated. The problem, though, is that many machine learning models built for this purpose do not work well because they are trained on limited data, and often the dataset itself is skewed — meaning there are far more healthy patients than sick ones. In this work, we tried to fix this by bringing together two ideas: synthetic data generation using CTGAN, and a stacking ensemble classifier. We first used CTGAN to produce new, realistic patient records that mirror the original data's patterns, then trained a stacking model — XGBoost, Random Forest, and Gradient Boosting as base learners, with Logistic Regression on top — on the combined real and synthetic dataset. When we tested it, the model hit 92% accuracy and beat the basic XGBoost model on every metric we checked, including precision, recall, and AUC. The takeaway is simple: adding synthetic data and stacking classifiers together noticeably strengthens cardiovascular risk prediction.

Keywords: Cardiovascular Disease, CTGAN, Synthetic Data Augmentation, Stacking Ensemble, Machine Learning

I. INTRODUCTION

Heart disease kills an enormous number of people globally every single year. When it is caught early, patients have a much better shot at recovery, and the overall cost of treatment also comes down. Over the past decade, electronic health records have become far more common, opening the door for machine learning tools to assist doctors in figuring out who is at risk.

That said, working with real clinical data is not straightforward. Most datasets are small, the number of sick patients is usually much lower than healthy ones, values are often missing, and sharing patient data between institutions is tightly restricted for privacy reasons. Taken together, these problems mean that a model trained on such data may perform well during development but poorly when used on new patients.

Our approach was to tackle these limitations directly. We used CTGAN to generate extra synthetic patient records to fill the data gap, and combined that with a stacking ensemble model to improve prediction reliability. The core idea is that if we feed the model richer, more balanced data, it should learn patterns that hold up better in practice.

II. LITERATURE SURVEY

Before building any machine learning pipeline, the data itself has to be put in order. Raw medical datasets almost always have gaps — missing entries, duplicate rows, inconsistent formats. Lee (2017) showed that cleaning and normalizing data before training noticeably cuts down noise and leads to better-performing models [1]. García et al. (2015) gave a thorough breakdown of techniques like encoding categorical columns and normalizing numeric ranges, which are standard preparation steps for data mining [2]. Alasadi and Bhaya (2017) made a similar point — that dropping duplicates and filling in missing values reduces bias and helps models learn more accurately [3]. All three of these works basically agree: if you skip preprocessing, the downstream model will suffer for it.

Once the data is clean, the next step is making sure the features given to the model are actually useful. Raw columns are not always the most informative representation. Zeng et al. (2015) proposed a selection method that picks features based on how they interact with each other, rather than just individually [4]. Rawat and Khemchandani (2017) showed how engineering new features from existing ones — ratios, combinations, and so on — can push classification accuracy higher [5]. Verdonck et al. (2024) argued that domain-based feature creation is increasingly important in modern applications

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

[6]. Nargesian et al. (2017) even automated this process, letting the model itself figure out which transformations help [7]. These papers make a strong case for investing time in feature design rather than feeding models raw columns as-is. Data augmentation has gained attention as a way to handle limited training data and class imbalance. Maharana et al. (2022) reviewed various augmentation strategies and found that more data variety leads to models that generalize better [8]. Chlap et al. (2021) focused on medical data and showed that augmented training sets improve model robustness and prediction ability [9]. Khosla and Saini (2020) found similar results — augmentation stabilizes training and makes the model less sensitive to which exact samples it sees during training [10].

In recent years, GANs have been applied to tabular data generation. Habibi et al. (2023) demonstrated that CTGAN can handle imbalanced tabular datasets and that the synthetic records it produces improve downstream model performance [11]. Ince et al. (2024) showed that CTGAN-generated datasets work well even in complex analytical settings [12]. Zhao et al. (2021) introduced CTAB-GAN, a variation that handles intricate feature correlations in tabular data more gracefully [14]. Fang et al. (2022) built DP-CTGAN, which generates synthetic medical records with formal privacy guarantees [15]. Majeed and Hwang (2023) used a similar GAN setup specifically for oversampling minority classes [16]. These studies collectively support the use of GAN-based methods for generating synthetic patient records.

Gradient boosting has been a go-to approach for structured data problems for years. Chen and Guestrin (2016) introduced XGBoost, which is fast, scalable, and typically outperforms simpler models on tabular data [17]. Chen et al. (2015) described the core algorithm in detail — it builds trees one after another, each correcting the errors of the last [18]. Sibindi et al. (2023) showed that combining different boosting methods can squeeze even more accuracy out of difficult datasets [22]. Bentéjac et al. (2021) ran a comparison study that put gradient boosting ahead of most alternatives on structured problems [24].

Where boosting reduces bias by focusing on hard examples, bagging reduces variance by training on different random subsets. Altman and Krzywinski (2017) explained how Random Forest combines many independently trained trees to produce a more stable prediction [19]. Ngo et al. (2022) extended this with evolutionary approaches to pick the best subset of trees [20]. Stacking goes one step further by learning how to best combine the outputs of different models. Rincy and Gupta (2020) found that ensemble methods, particularly stacking, consistently outperform individual models [25]. Mienye and Sun (2022) showed that a meta-learner trained on base model outputs tends to get the best of what each classifier offers [27].

III. PROBLEM STATEMENT

The core difficulty this project addresses is a practical one: there just is not enough good clinical data to train cardiovascular risk models reliably. Hospital datasets are small, often incomplete, and legally restricted in terms of sharing. On top of that, the number of heart disease patients in most datasets is far lower than the number of healthy patients, creating a class imbalance problem that hurts model training. The result is that models built on such data tend to memorize the training set rather than truly learn — and when they encounter new patients, their predictions fall apart.

IV. PROPOSED METHODOLOGY

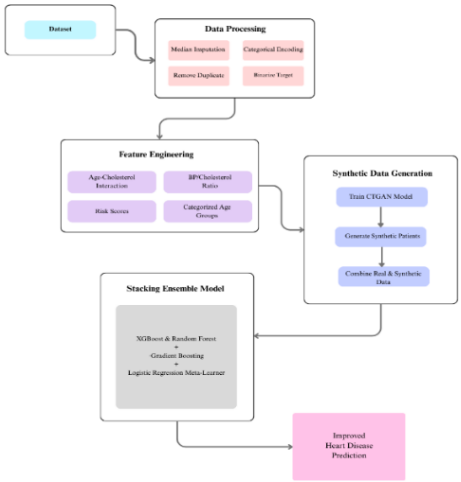

Figure 1. Overall Framework Architecture

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

A. Data Preprocessing

We started by cleaning the heart disease dataset. Where values were missing, we used median imputation rather than dropping those rows — dropping rows would have made the already small dataset even smaller. Duplicate entries were removed since they could push the model toward patterns that do not actually exist. Categorical columns were numerically encoded so the model could process them, and the target variable was mapped to binary (0 or 1). By the end of this step, the dataset was consistent and ready to work with.

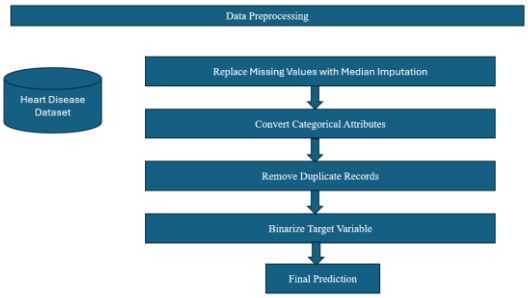

Figure 2. Data Preprocessing Pipeline

B. Feature Engineering

Raw features alone were not sufficient, so we created several new ones grounded in clinical reasoning. The ratio of blood pressure to cholesterol captures a combined risk signal that neither value alone would reveal. We also computed the interaction between maximum heart rate and age, and created combined age-cholesterol indicators. Patients were grouped into age brackets, and binary risk flags were set based on commonly used clinical thresholds. These engineered features gave the model more clinically meaningful signals to learn from.

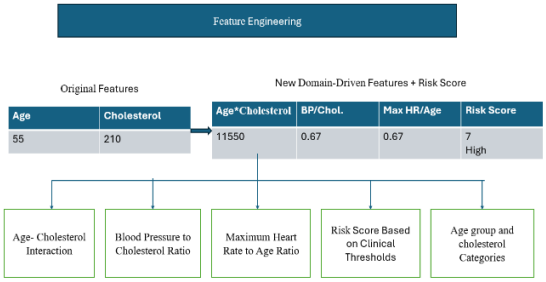

Figure 3. Feature Engineering Steps

C. Synthetic Data Generation Using CTGAN

To compensate for the limited number of patient records, we trained CTGAN on the preprocessed dataset for 300 epochs. CTGAN is designed specifically for tabular data — it handles mixed data types and tries to reproduce the statistical relationships between columns, not just the individual distributions. After training, we generated a batch of synthetic records, checked them for any obvious inconsistencies, corrected what we could, and then merged them with the original training data. The result was a much larger and more balanced training set.

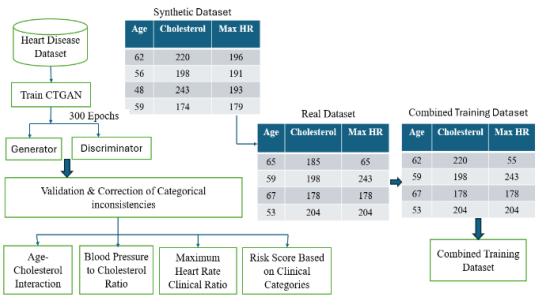

Figure 4. CTGAN Synthetic Data Generation Flow

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

D. Stacking Ensemble Model

With the augmented data ready, we built a stacking ensemble. Three models served as base learners — XGBoost, Random Forest, and Gradient Boosting — each chosen because it approaches classification differently. Their predictions were passed to a Logistic Regression meta-learner, which learned how to weight and combine them. Throughout training, we used cross-validation to ensure the meta-learner was not simply overfitting to the base learner outputs.

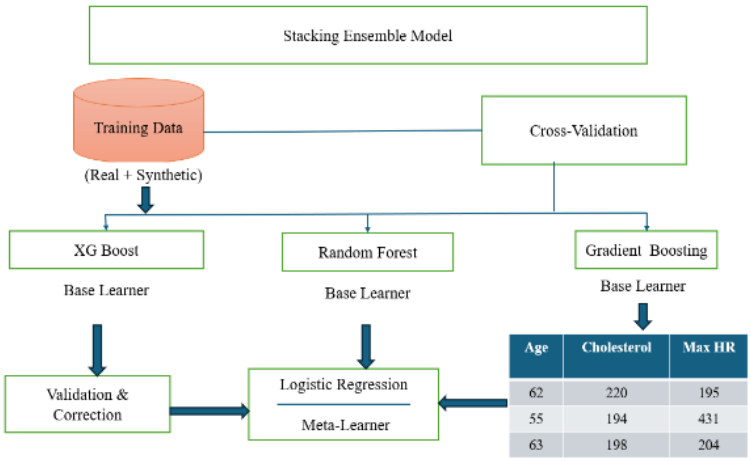

Figure 5. Stacking Ensemble Architecture

V. EXPERIMENTAL SETUP

A. Dataset Description

We used the UCI Heart Disease dataset for all experiments. It is a well-known benchmark in cardiovascular research. The dataset covers clinical attributes including age, blood pressure, cholesterol levels, maximum heart rate, resting ECG results, and other diagnostic indicators that are regularly used by clinicians when assessing heart disease risk. It is a compact dataset, which is precisely why augmentation was worth exploring here.

B. Data Split

We split the data 80/20 for training and testing. To ensure neither split ended up with a disproportionate number of heart disease cases, we used stratified sampling. This matters especially with smaller datasets where a random split could accidentally concentrate most of the positive cases in one set.

C. Evaluation Metrics

We evaluated the models using five metrics: Accuracy, Precision, Recall, F1-score, and AUC. Accuracy alone can be misleading with class imbalance, so Precision and Recall give a better picture of how the model handles the disease class specifically. F1-score balances the two, and AUC tells us how well the model separates the two classes regardless of threshold. For a medical application like this, Recall is especially critical — missing a true heart disease case is a much worse outcome than a false alarm.

VI. RESULTS AND DISCUSSION

A. Confusion Matrix Analysis

Table 1 shows results from the baseline XGBoost model trained only on real data.

Table 1. Baseline XGBoost Confusion Matrix

Actual \ Predicted	No Disease (0)	Disease (1)
No Disease (0)	26 (True Neg.)	7 (False Pos.)
Disease (1)	2 (False Neg.)	26 (True Pos.)

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

Table 2 shows the results after training on both real and synthetic data.

Table 2. Proposed Stacking Ensemble Confusion Matrix

Actual \ Predicted	No Disease (0)	Disease (1)
No Disease (0)	29 (True Neg.)	4 (False Pos.)
Disease (1)	1 (False Neg.)	27 (True Pos.)

B. Overall Performance Comparison

Table 3. Performance Comparison of Baseline vs. Proposed Model

Metric	Baseline	Proposed
Precision	0.86	0.92
Recall	0.86	0.92
F1-score	0.85	0.92
Accuracy	0.86	0.92

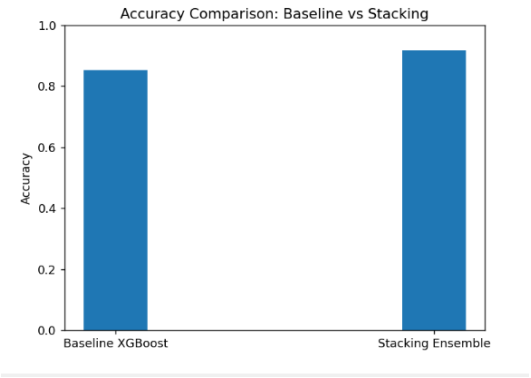

Figure 6. Overall Metric Comparison

The numbers tell a clear story. Across all four metrics, the stacking ensemble with synthetic augmentation beat the baseline. Accuracy went from 86% to 92% — a 6-point jump that, in a medical setting, translates to a meaningful number of patients being correctly classified. The reason is that the model was trained on a richer dataset, so it picked up on patterns that were invisible before due to data scarcity.

C. Class-wise Analysis

For the disease class (Class 1), precision improved from 0.79 (baseline) to 0.87 (proposed), as shown in Figure 7. A precision of 0.87 means that when the model flags someone as having heart disease, it is correct about 87% of the time.

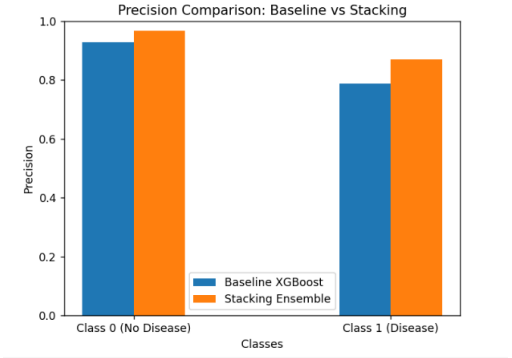

Figure 7. Precision Comparison by Class

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

Recall for the disease class went from 0.93 to 0.96, as illustrated in Figure 8. This is the metric that matters most from a clinical standpoint — it measures how many actual heart disease cases the model catches.

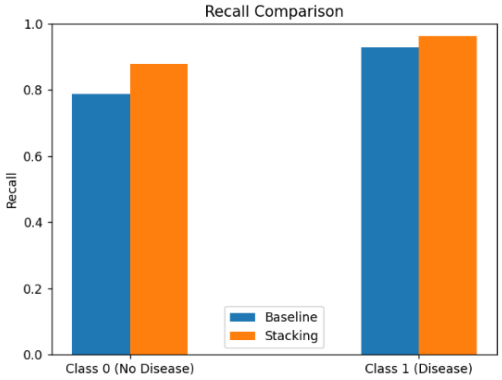

Figure 8. Recall Comparison by Class

The F1 improvements were consistent across both classes, as shown in Figure 9, which confirms that the gains were not achieved at the expense of one group.

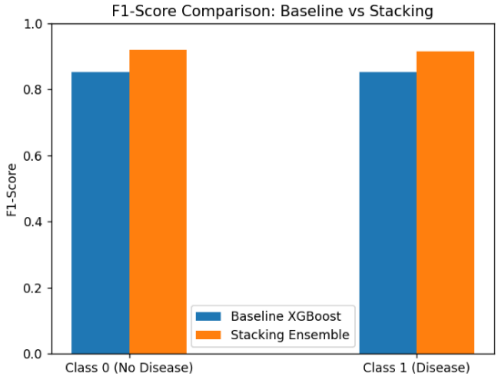

Figure 9. F1-Score Comparison by Class

D. ROC Curve Analysis

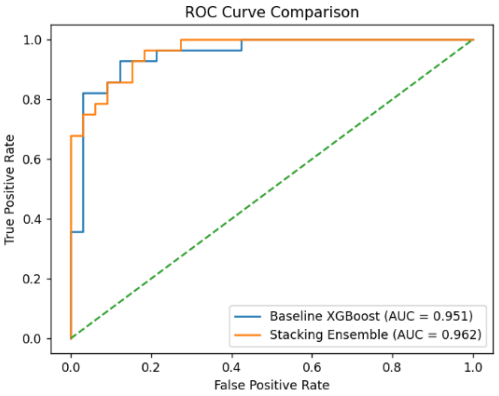

Figure 10. ROC Curve: Proposed Model vs. Baseline XGBoost

The ROC curves give another angle on model performance, showing how the True Positive Rate and False Positive Rate trade off at different decision thresholds. The baseline XGBoost model achieved an AUC of 0.951. The proposed stacking ensemble pushed that to 0.962. The proposed model's curve hugs the top-left corner more tightly, meaning it achieves high sensitivity without generating too many false alarms. Combining CTGAN-generated synthetic data with a stacking

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

ensemble thus improves the model's ability to discriminate between patients who have cardiovascular disease and those who do not.

VII. MATHEMATICAL FORMULATION

A. Problem Definition

Let $D = \{(x_i, y_i)\}$ for $i = 1$ to N be the dataset, where x_i is a d-dimensional feature vector for the i-th patient and $y_i \in \{0, 1\}$ is the class label (0 = no disease, 1 = disease present). The objective is to learn a mapping function $f: \mathbb{R}^d \rightarrow \{0, 1\}$ that correctly predicts the class label while minimizing prediction error.

B. CTGAN Objective Function

CTGAN frames data generation as a two-player minimax game between a generator G and a discriminator D:

$$min^G\ max^D\ V(D,G) = E[log\ D(x)] + E[log(1 - D(G(z)))]$$

Here p_data(x) is the distribution of real patient records and p_z(z) is random noise fed to the generator. Over time, G learns to match the real data distribution such that p_synthetic(x) ≈ p_data(x), making synthetic records statistically consistent with real ones.

C. Stacking Ensemble Model

Let $h_1(x)$, $h_2(x)$, and $h_3(x)$ be the predictions from XGBoost, Random Forest, and Gradient Boosting respectively. The meta-learner combines these via a weighted sum followed by a sigmoid activation:

$$H(x) = \sigma(w_1 \cdot h_1(x) + w_2 \cdot h_2(x) + w_3 \cdot h_3(x) + b)$$

The sigmoid function σ squashes the output to a probability. Weights w_i and bias b are learned during training so the meta-learner determines which base model to trust more in different situations.

D. Loss Function

Training uses Binary Cross-Entropy (BCE) loss:

$$L = -(1/N)\ \Sigma\ [\ y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)\ ]$$

where $\hat{y}_i$ is the predicted probability for the i-th sample. BCE penalizes confident wrong predictions more heavily, pushing the model toward well-calibrated probability estimates.

E. Evaluation Metrics

Five standard classification metrics were used:

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1 = 2 · Precision · Recall / (Precision + Recall)
AUC: Area under the ROC curve; measures class separability independent of threshold.

VIII. FINAL OUTPUT ANALYSIS

Looking at the two confusion matrices side by side is the clearest way to see what the proposed approach actually changed. The baseline XGBoost model gave: 26 true negatives, 7 false positives, 2 false negatives, and 26 true positives. The 2 missed cases are the bigger concern clinically — a delayed diagnosis of heart disease can have serious consequences.

After adding synthetic data and switching to the stacking ensemble, the numbers shifted to: 29 true negatives, 4 false positives, 1 false negative, and 27 true positives. We went from missing 2 patients to missing just 1 — a 50% reduction in the most dangerous error type. False positives also dropped from 7 to 4, meaning fewer patients are sent through unnecessary diagnostic procedures. The proposed framework performs better on both the metrics that matter most in this kind of application.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2026

DOI: 10.17148/IARJSET.2026.133100

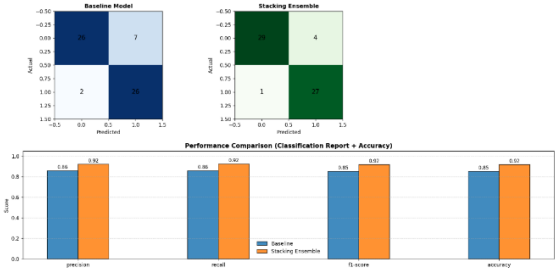

Figure 11. Final Confusion Matrix and Performance Summary

IX. CONCLUSION

This project set out to improve cardiovascular disease prediction by addressing two common and related problems: not enough training data, and poor handling of class imbalance. The solution we built uses CTGAN to generate synthetic patient records, then trains a stacking ensemble — XGBoost, Random Forest, and Gradient Boosting with Logistic Regression as the final layer — on the augmented dataset.

The results were encouraging. Compared to a standard XGBoost model trained on real data alone, the proposed system achieved higher accuracy (92% vs. 86%), precision, recall, and AUC. More importantly, it reduced false negatives — the cases where a patient with heart disease is incorrectly cleared. Cutting that by 50% is a meaningful outcome in a medical context. Feature engineering also played a larger role than initially expected; the clinically derived features gave the model context it would not have had from the raw columns alone.

Going forward, natural extensions include testing the framework on larger or multi-source datasets, experimenting with more advanced base learners, and looking at whether newer generative models can improve synthetic data quality. The current results suggest this approach is practically useful and could genuinely support early-stage cardiovascular risk screening in clinical settings.

A. Study Limitations

Several limitations should be noted. First, experiments were conducted on a single dataset (UCI Heart Disease), which limits generalizability. Second, the quality of CTGAN-generated data depends on the size and diversity of the original training set — with very small datasets, synthetic samples may not fully capture the real data distribution. Third, the stacking ensemble introduces additional computational complexity compared to a single classifier, which may be a consideration in resource-constrained clinical environments. Future work should validate the framework on multiple independent datasets and investigate the quality of synthetic data through domain expert review.

REFERENCES

[1]. Lee, M. (2017). Data preprocessing techniques in machine learning. Journal of Data Science, 7(3), 102-114.
[2]. García, S., Luengo, J., & Herrera, F. (2015). Data preprocessing in data mining (Vol. 72, pp. 59-139). Springer International Publishing.
[3]. Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences, 12(16), 4102-4107.
[4]. Zeng, Z., Zhang, H., Zhang, R., & Yin, C. (2015). A novel feature selection method considering feature interaction. Pattern Recognition, 48(8), 2656-2666.
[5]. Rawat, T., & Khemchandani, V. (2017). Feature engineering tools and techniques for better classification performance. International Journal of Innovations in Engineering and Technology, 8(2), 169-179.
[6]. Verdonck, T., Baesens, B., Óskarsdóttir, M., & vanden Broucke, S. (2024). Special issue on feature engineering editorial. Machine Learning, 113(7), 3917-3928.
[7]. Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. S. (2017). Learning feature engineering for classification. In IJCAI, 17, 2529-2535.
[8]. Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings, 3(1), 91-99.
[9]. Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. Journal of Medical Imaging and Radiation Oncology, 65(5), 545-563.

DOI: 10.17148/IARJSET.2026.133100

[10]. Khosla, C., & Saini, B. S. (2020). Enhancing performance of deep learning models with different data augmentation techniques: A survey. In ICIEM (pp. 79-85). IEEE.
[11]. Habibi, O., Chemmakha, M., & Lazaar, M. (2023). Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection. Engineering Applications of Artificial Intelligence, 118, 105669.
[12]. Ince, V., Bader-El-Den, M., & Sari, O. F. (2024). Enhanced dataset synthesis using CTGAN for metagenomic dataset. In IEEE 12th International Conference on Intelligent Systems (pp. 1-6). IEEE.
[13]. Ince, V., Bader-El-Den, M., & Sari, O. F. (2024). Enhanced dataset synthesis using CTGAN for metagenomic dataset. In IEEE 12th International Conference on Intelligent Systems (pp. 1-6). IEEE.
[14]. Zhao, Z., Kunar, A., Birke, R., & Chen, L. Y. (2021). CTAB-GAN: Effective table data synthesizing. In Asian Conference on Machine Learning (pp. 97-112). PMLR.
[15]. Fang, M. L., Dhami, D. S., & Kersting, K. (2022). DP-CTGAN: Differentially private medical data generation using CTGANs. In International Conference on AI in Medicine (pp. 178-188). Springer.
[16]. Majeed, A., & Hwang, S. O. (2023). CTGAN-MOS: Conditional GAN based minority-class-augmented oversampling for imbalanced problems. IEEE Access, 11, 85878-85899.
[17]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
[18]. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., & Zhou, T. (2015). XGBoost: Extreme gradient boosting. R package version 0.4-2, 1(4), 1-4.
[19]. Altman, N., & Krzywinski, M. (2017). Ensemble methods: Bagging and random forests. Nature Methods, 14(10), 933-935.
[20]. Ngo, G., Beard, R., & Chandra, R. (2022). Evolutionary bagging for ensemble learning. Neurocomputing, 510, 1-14.
[21]. Kadiyala, A., & Kumar, A. (2018). Applications of Python to evaluate the performance of bagging methods. Environmental Progress & Sustainable Energy, 37(5), 1555-1559.
[22]. Sibindi, R., Mwangi, R. W., & Waititu, A. G. (2023). A boosting ensemble learning based hybrid LightGBM and XGBoost model for predicting house prices. Engineering Reports, 5(4), e12599.
[23]. Malinin, A., Prokhorenkova, L., & Ustimenko, A. (2020). Uncertainty in gradient boosting via ensembles. arXiv preprint arXiv:2006.10562.
[24]. Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937-1967.
[25]. Rincy, T. N., & Gupta, R. (2020). Ensemble learning techniques and its efficiency in machine learning: A survey. In 2nd International Conference on Data, Engineering and Applications (pp. 1-6). IEEE.
[26]. Dasari, A. K., Biswas, S. K., Thounaojam, D. M., Devi, D., & Purkayastha, B. (2023). Ensemble learning techniques and their applications: An overview. In International Conference on Communications and Cyber Physical Engineering (pp. 897-912). Springer.
[27]. Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access, 10, 99129-99149.
[28]. M Kaliappan, E Mariappan, MV Prakash, B Paramasivan, Load Balanced Clustering Technique in MANET using Genetic Algorithms. Defence Science Journal 66 (3), 251-258.
[29]. M Sivaram, M Kaliappan, S J Shobana, Prakash, V Porkodi Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud, Journal of Ambient Intelligence and Humanized Computing, pp.1-9.
[30]. Vimal, S., Robinson, Y. H., Kaliappan, M., Vijayalakshmi, K., & Seo, S. (2021). A method of progression detection for glaucoma using K-means and the GLCM algorithm toward smart medical prediction. The Journal of Supercomputing, 77(1), 1-17. https://doi.org/10.1007/s11227-020-03268-0
[31]. Kaliappan M, Guruprakash B, Rajalakshmi, J. Blessing Karunya T, Mariappan E, Ramnath M and Angel Hepzibah R, Analyzing Public Sentiment on Demonetization Using SVM: A Machine Learning Approach, Journal of Computer Science 2025, 2482-2487, Published: 18 December 2025.

