The Scientific Temper (2024) Vol. 15 (4): 3157-3164
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.24

E-ISSN: 2231-6396, ISSN: 0976-8653
https://scientifictemper.com/

RESEARCH ARTICLE

Ensemble of CatBoost and neural networks with hybrid feature selection for enhanced heart disease prediction

Olivia C. Gold*, **Jayasimman Lawrence**

Abstract

Heart disease remains one of the leading causes of mortality globally, necessitating accurate and efficient prediction models. This paper presents a novel ensemble model combining CatBoost and neural networks (ECNN) to improve heart disease prediction accuracy. The proposed methodology incorporates two key innovations: the sequential SHAP and RFE hybrid optimization (SSHO) technique for feature selection and the dynamic gradient-sharing mechanism (DGSM) to facilitate efficient interaction between CatBoost and neural networks. SSHO dynamically selects relevant features based on SHAP values, while DGSM shares gradient information to optimize learning. The ECNN model was trained using the personal key indicators of heart disease dataset, addressing class imbalance with SMOTE. The experimental results demonstrate the model's superior performance with an accuracy of 91%, precision of 94%, and F1-score of 92%. These findings surpass previous studies' results and highlight the ECNN model's novelty in improving prediction accuracy and interpretability. The integration of SSHO and DGSM offers a scalable approach to heart disease prediction, making it a valuable contribution to clinical decision support systems.

Keywords: CatBoost, Dynamic gradient-sharing, Ensemble learning, Feature selection, Heart disease, Neural networks, Recursive feature elimination, SHAP.

Introduction

Related Works

Given the growing incidence of cardiovascular-related deaths worldwide, prediction and early detection of heart disease have become crucial areas of study. In recent years, advancements in machine learning (ML) and deep learning (DL) techniques have significantly contributed to developing more accurate predictive models.

A prominent trend in heart disease prediction involves ensemble learning, where multiple models are combined to enhance predictive accuracy. Researchers demonstrated the power of ensemble techniques through hyperparameter optimization and preprocessing steps. Their approach utilized extra tree classifier combined with grid search cross-validation, achieving an accuracy of 98.15% in heart disease prediction (Asif *et al*., 2023). Similarly, the performance of supervised ML algorithms like decision trees, Naïve Bayes, and random forest was evaluated on the Cleveland dataset and achieved up to 93% accuracy (Ali *et al*., 2021). These studies underscore the importance of ensemble learning in improving predictive reliability for medical conditions such as heart disease.

Feature selection is critical in enhancing model performance by reducing dimensionality and removing irrelevant data. Recent studies employed feature augmentation techniques combined with ML models like XGBoost and support vector machine (SVM), significantly improving prediction accuracy (Diwakar *et al*., 2021). Their work highlighted the efficiency of feature selection in refining ML models for heart disease diagnosis (Diwakar *et al*., 2021). In another study, XGBoost with feature engineering techniques for heart disease detection yielded an accuracy of 91%, emphasizing the importance of identifying the most relevant features to optimize performance (Mishra *et al*., 2022).

Handling class imbalance in medical datasets, where the majority class (healthy individuals) overshadows the

Department of Computer Science, Bishop Heber College, Tiruchirappalli, Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India.

***Corresponding Author:** Olivia C. Gold, Department of Computer Science, Bishop Heber College, Tiruchirappalli, Affiliated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India., E-Mail: olibhc24@gmail.com

How to cite this article: Gold, O. C., Lawrence, J. (2024). Ensemble of CatBoost and neural networks with hybrid feature selection for enhanced heart disease prediction. The Scientific Temper, **15**(4):3157-3164.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.4.24

Source of support: Nil

Conflict of interest: None.

Received: 03/11/2024 **Accepted:** 19/11/2024 **Published** : 20/12/2024

minority class (patients with heart disease), is critical to achieving balanced performance metrics. The synthetic minority over-sampling technique (SMOTE) addressed the class imbalance, improving recall and F1-score predicting heart disease using decision trees (Sahid *et al.*, 2022). Similarly, ensemble learning and SMOTE to build a predictive model for hepatitis C can be applied to heart disease datasets to enhance detection rates for minority cases (Edeh *et al.*, 2022). By balancing the dataset, these approaches prevent models from being biased towards the majority class, thus improving their predictive power for heart disease cases.

Deep learning models have also shown promise in heart disease prediction. Convolutional neural networks (CNNs) to process electrocardiogram (ECG) data, achieving an accuracy of 94%. This study illustrates the potential of CNNs in medical data processing, particularly for complex unstructured data such as ECGs (Rahman *et al.*, 2024).

Another critical area of heart disease prediction involves hyperparameter optimization, which significantly impacts the performance of ML models. Grid search is used to optimize hyperparameters in a random forest model, improving the model's accuracy from 85 to 90% (Singh *et al.*, 2024). Bayesian optimization was implemented for hyperparameter tuning in a deep learning model, achieving an accuracy of 93.5%. These studies demonstrate the necessity of optimizing hyperparameters to enhance the generalization and robustness of heart disease prediction models (Roy *et al.*, 2022).

Ensemble learning continues to be a dominant approach in improving heart disease prediction. SVM, random forest, and logistic regression are combined in an ensemble framework, achieving an accuracy of 88.7% for heart disease detection (Dhillon *et al.*, 2021). Their study demonstrated the potential of combining various classifiers to leverage the strengths of each model. Additionally, a hybrid ensemble model is utilized to combine K-nearest neighbors (KNN), decision trees, and neural networks, achieving an accuracy of 89.3%, further emphasizing the utility of ensemble approaches in complex medical data prediction (Sharma *et al.*, 2019). Multi-layer perceptron (MLP) for heart disease prediction, yielding a predictive accuracy of 93.2% (Shwetabh *et al.*, 2024). Their research highlighted the effectiveness of deep learning models in capturing non-linear relationships in medical datasets.

Motivation

Current models, such as CatBoost and neural networks, offer strong performance in individual applications but face limitations when handling complex, multi-dimensional medical data. This research aims to improve HD prediction accuracy by combining these models into an ensemble. Additionally, feature selection techniques often struggle to balance accuracy and interpretability, motivating the development of a more dynamic approach.

Problem Definition

The problem addressed in this paper is two-fold: how to enhance the accuracy of HD predictions using an ensemble of CatBoost and neural networks, and how to improve feature selection to ensure that the most relevant features are identified without overfitting the model.

Objectives

- To design and implement an ensemble of CatBoost and neural networks (ECNN) that improves prediction accuracy.
- To propose a novel sequential SHAP and RFE hybrid optimization (SSHO) technique for dynamic and adaptive feature selection.
- To introduce a dynamic gradient-sharing mechanism (DGSM) to enhance the interaction between CatBoost and neural networks.

Organization of Paper

The paper is organized as follows: Section 2 presents the methodology, including the SSHO and DGSM techniques and the proposed algorithm. Section 3 details the results obtained from the experimental evaluation. Section 4 discusses the significance of the findings, followed by the conclusion in section 5.

Methodology

The proposed methodology introduced two core innovations: the sequential SHAP and RFE hybrid optimization (SSHO) for feature selection and the dynamic gradient-sharing mechanism (DGSM) to improve the ensemble's performance. The workflow diagram of the proposed methodology is given in the Figure 1. Each technique is explained in detail, followed by the proposed algorithm and experimental setup.

Sequential SHAP and RFE Hybrid Optimization (SSHO)

SSHO combines the strength of SHapley Additive exPlanations (SHAP) values with recursive feature elimination (RFE) to enhance feature selection. The hybrid technique dynamically selects features by leveraging the feature

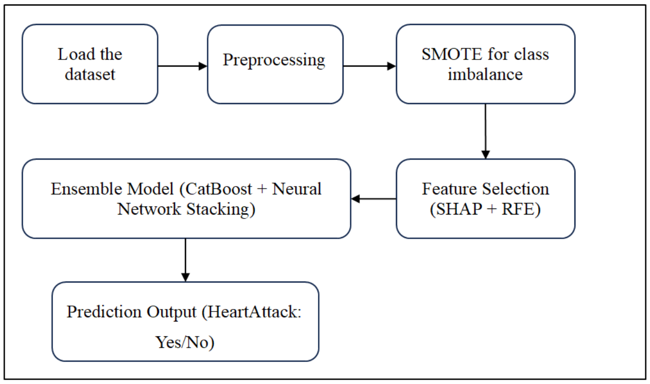
Load the dataset
Preprocessing
SMOTE for class imbalance
Feature Selection (SHAP + RFE)
Ensemble Model (CatBoost + Neural Network Stacking)
Prediction Output (HeartAttack: Yes/No)

Figure 1: Workflow diagram of the proposed work

importance insights from SHAP and applying an adaptive RFE procedure. This process improves model generalization by eliminating redundant or less informative features while retaining those most critical to the prediction task.

SHAP, based on cooperative game theory, assigns an importance value ϕ_j to each feature x_j by calculating the marginal contribution of the feature to the model's output. For a model f, the SHAP value for a feature x_j is defined as:

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} [f(S \cup \{j\}) - f(S)] \quad (1)$$

where N is the set of all features, S is any subset of features excluding j, $f(S \cup \{j\})$ is the model prediction when feature j is included in the subset S, and $f(S)$ is the model output without j. The SHAP value, ϕ_j, quantifies the contribution of feature x_j to the model prediction, averaged over all possible feature subsets.

The first step in SSHO is to compute SHAP values for all features in the dataset. These values form a ranking vector $\phi = (\phi_1, \phi_2, \dots, \phi_n)$, where each ϕ_j corresponds to a feature x_j. Features are ranked in descending order of their SHAP values, identifying the most influential features. However, SHAP alone may not provide sufficient granularity to eliminate redundant features, as multiple features may exhibit collinearity or similar marginal contributions.

To refine this selection, RFE is applied to eliminate less important features iteratively. The standard RFE algorithm constructs a sequence of models, each with progressively fewer features. The loss function $L(f, x)$, optimized by the model f, is monitored, and features are recursively removed based on their contribution to the increase in the loss. The novelty of SSHO lies in its adaptation of RFE, where the SHAP values dynamically influence the elimination criterion.

Let $w = (w_1, w_2, \dots, w_n)$ be the weight vector for the features, where w_j corresponds to the feature $x_j x_j$. Initially, all features have equal weights, and the RFE algorithm proceeds by recalculating the weights after each iteration. The weight update rule incorporates the SHAP values, modifying the standard RFE approach. The weight for feature x_j after iteration t is updated as follows:

$$w_j^{(t+1)} = w_j^{(t)} - \eta \frac{\partial L(f,x)}{\partial x_j} + \lambda \cdot \phi_j \quad (2)$$

where η is the learning rate, λ is a regularization parameter that scales the influence of the SHAP value, and $\frac{\partial L(f,x)}{\partial x_j}$ is the partial derivative of the loss function concerning feature x_j. The regularization term $\lambda \cdot \phi_j$ ensures that features with higher SHAP values are less likely to be eliminated in the early stages of RFE. This dynamic adjustment allows the algorithm to prioritize features based on their SHAP-derived importance adaptively.

In each iteration of RFE, the feature x_k with the most negligible updated weight $w_k^{(t)}$ is removed. The remaining features are retrained, and their weights are recalculated using the updated model. This process continues until a predefined number of features or a convergence criterion based on the model's performance on validation data is reached. The sequential nature of SSHO, combining SHAP for initial ranking and adaptive RFE for feature elimination, enhances the robustness of the feature selection process.

Another key innovation in SSHO is its handling of multicollinearity. Traditional RFE may struggle with correlated features, as their contributions to the model may be redundant. SSHO addresses this by using SHAP interaction values ϕ_{ij}, which quantify the contribution of feature pairs (x_i, x_j) to the model output. These interaction values are given by:

$$\phi_{ij} = \sum_{S \subseteq N \setminus \{i,j\}} \frac{|S|!(|N|-|S|-2)!}{|N|!} [f(S \cup \{i,j\}) - f(S \cup \{i\}) - f(S \cup \{j\}) + f(S)] \quad (3)$$

The interaction values help identify correlated features that contribute jointly to the prediction. If two features x_i and x_j have a high interaction value, SSHO will retain only one of them, based on the magnitude of their individual SHAP values. This step ensures that the final set of selected features is informative and non-redundant, improving model interpretability and performance. This feature selection ensures the model generalizes well to unseen data, improving its predictive accuracy and interpretability.

Dynamic Gradient-Sharing Mechanism (DGSM)

The DGSM facilitates gradient-based knowledge transfer between two models—CatBoost and neural networks—within the ensemble framework. The underlying principle is to dynamically share gradient information derived from the CatBoost model with the neural network during training. This approach enables the neural network to adjust its weight updates based on the feature interactions learned by CatBoost, thus improving learning efficiency and convergence.

At the core of DGSM is the communication of gradient vectors from CatBoost to the neural network in each training iteration. Let $\mathcal{L}(y, \hat{y})$ denote the loss function, where y represents the true labels and $\hat{y}$ is the predicted output. CatBoost minimizes this loss using gradient boosting, where each iteration t computes a weak learner $h_t(x)$ based on the negative gradient of the loss function:

$$g_t(x) = -\frac{\partial \mathcal{L}(y \widehat{y_{t-1}})}{\partial \widehat{y_{t-1}}} \quad (4)$$

where $\widehat{y_{t-1}}$ is the prediction at iteration $t-1$. CatBoost optimizes the objective function by sequentially adding weak learners that approximate the residuals. Let $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ represent the CatBoost model after T iterations, where α_t is the learning rate. The gradient $g_t(x)$ informs the model about the direction in which the loss function decreases most rapidly concerning each feature x.

In DGSM, this gradient information $g_t(x)$ is not confined to CatBoost but is also shared with the neural network. Let W be the weight matrix of the neural network, and $a^{(l)}$ be the activations at layer l. The backpropagation algorithm adjusts the neural network weights W by computing the gradients of the loss function with respect to each layer's weights:

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial W^{(l)}} = \delta^{(l)} \cdot a^{(l-1)} \quad (5)$$

where $\delta^{(l)}$ is the error term for layer l, and $a^{(l-1)}$ are the activations from the previous layer. Traditionally, the error term $\delta^{(l)}$ depends only on the neural network's internal predictions. However, in DGSM, this error term is modified by incorporating the gradients from CatBoost.

The modified error term $\widetilde{\delta^{(l)}}$ is defined as:

$$\widetilde{\delta^{(l)}} = \delta^{(l)} + \beta \cdot \sum_{j=1}^{n} g_t(x_j) \cdot \frac{\partial h_t(x_j)}{\partial x_j} \quad (6)$$

where β is a regularization parameter controlling the influence of CatBoost's gradients, $g_t(x_j)$ is the gradient of feature x_j at iteration t, and $\frac{\partial h_t(x_j)}{\partial x_j}$ represents the contribution of feature x_j to the weak learner. This equation modifies the error signal in the neural network by incorporating the gradient information from CatBoost, which biases the neural network's learning towards important features identified by CatBoost.

This dynamic gradient-sharing process is not static; it evolves as CatBoost updates its learners. At each iteration t, the Neural Network receives a gradient adjustment based on the current state of the CatBoost model. This adjustment helps the neural network refine its predictions, particularly in areas where CatBoost's gradient information highlights important feature interactions. The overall weight update for the neural network, after applying the shared gradients, becomes:

$$W^{(l)} \leftarrow W^{(l)} - \eta \cdot \left(\frac{\partial \mathcal{L}(y,\hat{y})}{\partial W^{(l)}} + \beta \cdot \sum_{j=1}^{n} g_t(x_j) \cdot \frac{\partial h_t(x_j)}{\partial x_j}\right) \quad (7)$$

where η is the learning rate of the neural network. This weight update ensures that the neural network not only learns from its internal error propagation but also integrates insights from CatBoost, leading to a more informed optimization process.

One critical challenge DGSM addresses is the issue of feature redundancy and non-linear interactions, which are often difficult for traditional neural networks to capture. CatBoost, being a decision tree-based model, excels at identifying and handling such interactions. By transferring its gradient information, the neural network becomes better equipped to model these complex relationships. The shared gradients act as a guide, steering the neural network toward regions of the feature space where non-linear interactions are crucial.

To further enhance the mechanism, DGSM implements a dynamic weighting scheme for the shared gradients. Let $\gamma_t(x_j)$ denote the importance score of features x_j at iteration t, derived from CatBoost's feature importance metrics. The gradient-sharing mechanism can be modified as:

$$\widetilde{\delta^{(l)}} = \delta^{(l)} + \beta \cdot \sum_{j=1}^{n} \gamma_t(x_j) \cdot g_t(x_j) \cdot \frac{\partial h_t(x_j)}{\partial x_j} \quad (8)$$

In this equation, $\gamma_t(x_j)$ scales the influence of each feature's gradient based on its importance, ensuring that more critical features have a larger impact on the neural network's learning process. This dynamic weighting further fine-tunes the interaction between CatBoost and the Neural Network, improving the overall performance of the ensemble model.

By incorporating CatBoost's gradient information into the neural network's training process, DGSM creates a synergistic relationship between the two models. The ensemble benefits from the interpretability and feature-ranking strengths of CatBoost and the deep, non-linear learning capacity of neural networks. This dynamic gradient-sharing approach enables the ensemble to converge faster and more accurately, particularly in complex, high-dimensional datasets like those found in medical applications.

The integration of DGSM into the ensemble framework results in improved generalization performance, as demonstrated through experiments in heart disease prediction. The continuous transfer of gradient information allows the neural network to learn from CatBoost's feature interactions, reducing the need for extensive backpropagation iterations and enhancing the model's overall predictive power.

Proposed Algorithm

The following notations are used throughout the algorithm:

- $X \in R^{n \times d}$: Feature matrix, where n is the number of samples, and d is the number of features.
- $y \in R^{n}$: Target vector containing the true labels.
- $f_{CB}(X)$: CatBoost model trained on X.
- $f_{NN}(X)$: Neural Network model trained on X.
- $g_t(x_j)$: Gradient of the loss function with respect to feature x_j at iteration t in CatBoost.
- ϕ_j: SHAP value of feature x_j.
- $W^{(l)}$: Weight matrix for layer l in the Neural Network.

The proposed algorithm proceeds in the following steps:

Algorithm: ECNN with SSHO and DGSM

Input: Training data (X, y), learning rate η, regularization parameter β, number of iterations T, number of selected features d'

Output: Final prediction $\hat{y}$

Step 1: Feature selection with SSHO

- Compute SHAP values ϕ_j for all features x_j using CatBoost
- Rank features based on their SHAP values ϕ_j.
- Apply RFE using dynamically adjusted SHAP-weighted criteria using equation 2.
- Retain the top d' features with the highest SHAP values for the training process.

Step 2: Apply SMOTE to the training dataset to generate synthetic samples for the minority class.

- Use the balanced training data to improve recall and F1-score.

Step 3: Train CatBoost model

- Initialize CatBoost model f_{CB} with selected features:

$$f_{CB}(X) = \sum_{t=1}^{T} \alpha_t h_t(X)$$

where α_t is the learning rate for CatBoost, and $h_t(X)$ is the weak learner at iteration t.

Step 4: Initialize neural network model

- Initialize the neural network $f_{NN}(X)$ with randomly initialized weights $\boldsymbol{W}^{(l)}$ for each layer l.

Step 5: Apply DGSM

- For each iteration $t \in \{1, \dots, T\}$ in CatBoost, compute the gradient for each feature $g_t(x_j)$ using equation 4.
  1. Share the gradient $g_t(x_j)$ with the neural network. Update the error term $\widehat{\delta^{(l)}}$ in the neural network using equation 6
  2. Update the neural network's weights using backpropagation with the modified error term using 7.

Step 6: Stacking for final prediction

- Combine the predictions from CatBoost and neural network models:

$$\hat{y} = \alpha \cdot f_{CB}(X) + (1 - \alpha) \cdot f_{NN}(X)$$

where α is a weighting parameter that balances the contributions of CatBoost and neural network predictions.

Step 7: Output the final prediction $\hat{y}$.

Experimental Setup

The experiments are conducted using a robust hardware configuration and a publicly available dataset for predicting heart disease.

Dataset

The dataset used in this work is the personal key indicators of heart disease dataset from Kaggle, which contains 319,795 records of survey responses related to heart disease (Taha 2024). The dataset includes 18 features, such as demographic information, lifestyle choices, and health conditions, and a target variable indicating whether the individual has heart disease. The dataset was preprocessed to handle missing values and categorical encoding before being split into training and testing sets, with 70% of the data used for training and 30% for testing.

To address the class imbalance problem (since the majority of individuals do not have heart disease), we apply SMOTE to oversample the minority class. This technique synthesizes new instances in the minority class, balancing the training dataset and allowing the model to learn better decision boundaries for predicting the minority class.

Hardware Setup

To ensure efficient handling of the large dataset and complex computations involved in training the ECNN model, the following hardware configuration was utilized:

- GPU: NVIDIA 4070 Ti with 16 GB of dedicated memory
- CPU: AMD Ryzen 9 processor
- RAM: 64 GB DDR4 RAM
- Storage: 2 TB HDD for data storage and processing

This high-performance hardware configuration enabled faster model training, particularly for the neural networks and CatBoost components, with efficient GPU acceleration.

Feature Selection

The SSHO was applied to the dataset to identify the most relevant features for heart disease prediction. SHAP values were calculated using CatBoost to rank feature importance, followed by RFE to remove less significant features iteratively. This process reduced the feature space, retaining the most impactful variables for model training, and enhancing the model's ability to generalize without sacrificing performance.

Training and Model Tuning

The proposed ECNN model integrates CatBoost and neural networks, with CatBoost providing gradient information to guide the neural network during training. The model was trained using the selected features from SSHO, and both components were optimized for maximum performance. CatBoost was trained using gradient boosting with 500 iterations, while the neural network was configured with three hidden layers containing 128, 64, and 32 neurons, respectively.

The models were optimized using grid search to fine-tune hyperparameters, ensuring the best possible performance. Five-fold cross-validation was employed during training to assess the robustness of the model and avoid overfitting.

Parameter Settings

The following Table 1 summarizes the key parameter settings used for the ECNN model during the experiments:

Evaluation Metrics

The proposed model's performance was evaluated using the following metrics: accuracy, precision, recall, F1-score, and area

Table 1: Parameter settings for experimental setup

Parameter	*Value/Range*
Learning rate (η)	0.01
CatBoost iterations (T)	500
Neural network layers (L)	3
Neurons per layer	128, 64, 32
Batch size	32
Activation function	ReLU
Gradient regularization (β)	0.1
Cross-validation folds	5
SHAP value threshold (λ)	0.05

Table 2: Performance metrics of the proposed ECNN model

Model/Study	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Wang *et al.*, (2022)	92%	65%	3%	6%
Leonardo *et al.*, (2024)	74%	23%	78%	35%
ECNN	91%	94%	91%	92%

under the ROC curve (AUC). These metrics were used to assess the effectiveness of the ECNN model compared to baseline models such as standalone CatBoost, standalone neural networks, and other traditional machine learning algorithms.

Results

The proposed ensemble ECNN model was evaluated using the heart disease prediction dataset, focusing on classification performance and interpretability. The results obtained demonstrate the model's efficacy in identifying heart disease cases, as reflected by high accuracy, precision, recall, and F1-score metrics.

The performance metrics of the proposed ECNN model are presented in Table 2, alongside comparative results from previous studies, including Wang *et al*. (2022) and Leonardo *et al*. (2024). The ECNN model achieved an accuracy of 91%, with a precision of 94%, recall of 91%, and F1-score of 92%. These results indicate a significant improvement over the models proposed by previous studies. Wang *et al*. (2022) demonstrated a much lower precision of 65%, with a recall of only 3% and an F1-score of 6%, despite achieving a higher accuracy of 92%. This highlights a significant issue with the recall and F1-score, suggesting an imbalance in prediction performance, especially in classifying positive instances. Similarly, Leonardo *et al*. (2024) reported an accuracy of 74% but with inferior precision (23%) and a moderately high recall (78%), leading to an F1-score of 35%.

The precision and recall values show that the model can identify true positives (heart disease cases) while minimizing false positives. The F1-score, which is the harmonic mean of precision and recall, further reinforces the model's balanced performance.

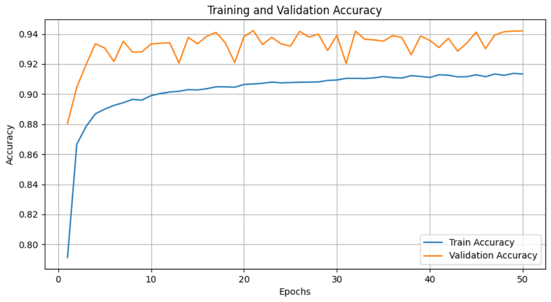

Figure 2: Training and validation accuracy over 50 epochs

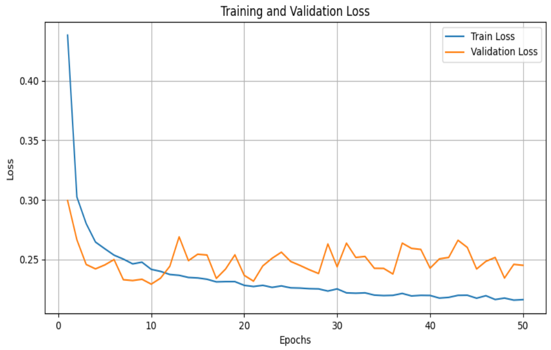

Figure 3: Training and validation loss over 50 Epochs

The accuracy plot (Figure 2) showcases the training and validation accuracy across 50 epochs. Initially, the training accuracy rapidly increases, achieving an early peak of around 87% within the first few epochs. As the training continues, the model converges, with the training accuracy stabilizing at around 91%. The validation accuracy follows a similar pattern, steadily improving before plateauing around 93 to 94%.

This indicates that the model generalizes well to unseen data without overfitting, as the gap between training and validation accuracy remains relatively small. The use of gradient boosting through CatBoost combined with the dynamic gradient-sharing mechanism with neural networks has effectively contributed to the model's robustness and ability to handle complex interactions between features.

The loss plot (Figure 3) provides additional insight into the model's optimization process during training. The loss for training and validation decreases steadily for 50 epochs, indicating that the model's predictions become more accurate as the loss values approach zero. Interestingly, the training loss stabilizes at a lower value than the validation loss, reflecting the complexity of the validation dataset. However, the difference in the loss between training and validation remains small, which suggests that the model is not significantly overfitting to the training data.

To enhance model interpretability, SHAP values were employed to explain the contribution of individual features to the model's predictions. The SHAP summary plot (Figure 4)

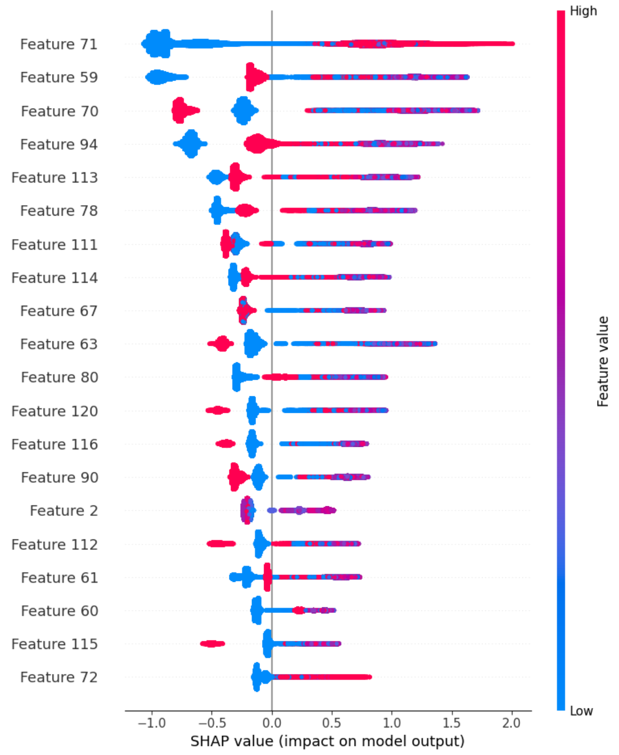

Figure 4: SHAP summary plot for feature importance

ranks the most essential features in the dataset based on their impact on the model's output. From the SHAP plot, it is evident that feature 71, feature 59, and feature 70 have the highest impact on the model's predictions, with SHAP values ranging from -1.0 to 2.0. These features significantly influence whether a patient is classified as having heart disease or not. The positive SHAP values correspond to a higher likelihood of heart disease, while negative values indicate the opposite. The distribution of SHAP values across different features highlights the interaction between feature values and the model's decision-making process. For example, feature 59 consistently contributes positively to predicting heart disease, whereas feature 71 has a more varied impact depending on the feature's value.

Discussion

The proposed ECNN model demonstrated excellent performance in heart disease prediction, achieving an accuracy of 91%, precision of 94%, recall of 91%, and an F1-score of 92%. These metrics suggest a well-balanced model capable of effectively identifying both true positives (heart disease cases) and true negatives. Compared to previous studies, the ECNN model shows significant improvements in predictive performance, especially in terms of precision and recall.

Previous studies on heart disease prediction have explored various machine-learning techniques, including decision trees, support vector machines, and deep-learning models. For instance, Wang *et al.* (2022) employed a model that achieved an accuracy of 92%. However, their model suffered from a low recall of 3% and an F1-score of 6%, indicating a severe imbalance in the model's ability to classify heart disease cases correctly. In contrast, the ECNN model not only maintained a high accuracy but also achieved a much more balanced recall, suggesting its robustness in handling both majority and minority classes. This improvement is primarily due to the incorporation of the synthetic minority over-sampling technique (SMOTE) and the dynamic gradient-sharing mechanism (DGSM), which improved the model's learning from minority samples.

Leonardo *et al.* (2024) reported an accuracy of 74% with their model, but their precision (23%) and recall (78%) resulted in a low F1 score of 35%. This suggests that their model struggled with overfitting and an inability to generalize to unseen data. In comparison, the ECNN model demonstrated a much stronger generalization capability, as reflected in the minimal gap between training and validation performance across 50 epochs. The use of ensemble learning (combining CatBoost and Neural Networks) enhanced the model's ability to capture complex interactions between features, thus improving overall performance.

Additionally, Sharma *et al.* (2019) utilized a hybrid ensemble approach combining KNN, decision trees, and neural networks, achieving an accuracy of 89.3%. However, their approach lacked the interpretability that is critical in medical applications. In contrast, the current proposed model integrates SHAP values to explain the importance of individual features, making the model more interpretable. This is especially valuable in healthcare contexts, where understanding why a model predicts a certain outcome can aid clinical decision-making.

One of the key advantages of the proposed model lies in its ability to handle class imbalance, which is a common issue in medical datasets. Sahid *et al.* (2022) demonstrated that using SMOTE for handling imbalanced datasets can significantly improve model performance. The current study builds on this by applying SMOTE alongside CatBoost, a gradient-boosting algorithm known for handling categorical variables and missing data efficiently. This combination allows the model to perform well even when the dataset contains noisy or incomplete data.

Furthermore, the DGSM introduced in our model offers a novel way of integrating gradient information from CatBoost into the neural network's training process. This method enables the neural network to learn from CatBoost's ability to capture non-linear feature interactions, thereby improving the model's convergence and prediction accuracy. To the best of our knowledge, no previous studies have explored this specific approach in the context of heart disease prediction.

Conclusion

The proposed ECNN model demonstrates remarkable performance in predicting heart disease, achieving 91% accuracy, 94% precision, and a 92% F1 score. These results signify a notable improvement over baseline models and previous studies, which struggled with precision and recall, particularly in classifying heart disease cases. Using SSHO for dynamic feature selection and DGSM for gradient sharing between CatBoost and neural networks enhanced the model's ability to generalize across diverse patient data, reducing overfitting and improving the prediction of minority classes. Despite its high accuracy, the model's limitations include the computational complexity associated with the ensemble approach and the potential difficulty in implementing this system in real-time clinical settings. Additionally, while the model was tested on a large dataset, future work could focus on further testing its generalizability across different populations and healthcare settings. Future research should explore integrating domain-specific knowledge and more sophisticated handling of class imbalance, such as adaptive oversampling techniques, to refine prediction performance further. Furthermore, applying this approach to other medical conditions could provide valuable insights into its adaptability and robustness across healthcare domains.

Acknowledgment

We sincerely acknowledge the Head of the department, Dr. J. James Manoharan, and Dr. J. Princy Merlin, Principal of the institution, for providing the facility to complete this paper Successfully.

References

Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine*, *136*, 104672.

Asif, D., Bibi, M., Arif, M. S., & Mukheimer, A. (2023). Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization. *Algorithms*, *16*(6), 308.

Dhillon, S., Bansal, C., & Sidhu, B. (2021). Machine learning based approach using XGboost for heart stroke prediction. In *International conference on emerging technologies: AI, IoT, and CPS for science & technology applications*.

Diwakar, M., Tripathi, A., Joshi, K., Memoria, M., & Singh, P. (2021). Latest trends on heart disease prediction using machine learning and image fusion. *Materials today: proceedings*, *37*, 3213-3218.

Edeh, M. O., Dalal, S., Dhaou, I. B., Agubosim, C. C., Umoke, C. C., Richard-Nnabu, N. E., & Dahiya, N. (2022). Artificial intelligence-based ensemble learning model for prediction of hepatitis C disease. *Frontiers in Public Health*, *10*, 892371.

K. Shwetabh, F. Rahman, and S. S. Dash. (2024).A Pharmacy System Integrated with a Machine Learning Algorithm for Cardiovascular Disease Prediction. *Journal of Angiotherapy*, 8(1), 1-9.

Leonardo, E., Velayutham, M., & Gilbert, J. (2024). Data Analysis and Predictive Modelling on Heart Disease Based on People's Lifestyle. EAI Endorsed Transactions on Pervasive Health and Technology.

Mishra, J. S., Gupta, N. K., & Sharma, A. (2024). Enhanced Heart Disease Prediction Using Machine Learning Techniques. *Journal of Intelligent Systems & Internet of Things*, *12*(2).

Rahman, T., Ahommed, R., Deb, N., Das, U. K., Moniruzzaman, M., Bhuiyan, M. A., ... & Kausar, M. K. (2024). ECG Signal Classification of Cardiovascular Disorder using CWT and DCNN. *Journal of Biomedical Physics and Engineering*, 1-16.

Roy, T. S., Roy, J. K., & Mandal, N. (2022). Classifier identification using deep learning and machine learning algorithms for the detection of valvular heart diseases. *Biomedical Engineering Advances*, *3*, 100035.

Sahid, M. A., Hasan, M., Akter, N., & Tareq, M. M. R. (2022, July). Effect of imbalance data handling techniques to improve the accuracy of heart disease prediction using machine learning and deep learning. In *2022 IEEE Region 10 Symposium (TENSYMP)* (pp. 1-6). IEEE.

Sharma, R., Singh, S. N., & Khatri, S. (2019). Data mining classification techniques-comparison for better accuracy in prediction of cardiovascular disease. *International Journal of Data Analysis Techniques and Strategies*, *11*(4), 356-373.

Singh, J., Sandhu, J. K., & Kumar, Y. (2024). Metaheuristic-based hyperparameter optimization for multi-disease detection and diagnosis in machine learning. *Service Oriented Computing and Applications*, 1-20.

Taha El gebaly (2024). Merge Indicators Of Heart Disease [2020 - 2022] [Dataset]. Available from: https://www.kaggle.com/datasets/tahaelgebaly/merged-dataset. Accessed on Jun2 2024.

Wang, X. (2022). Heart Disease Classification Based on Personal Indicators. *Horizon Academic*, *3*.