UTRECHT UNIVERSITY

Predicting Heart Disease Diagnosis: A Custom-Built Neural Network Versus Classical Machine Learning Models

Bachelor Thesis Artificial Intelligence
7.5 EC

Andreas Meeldijk
0892734

Supervisor: Baharak Ibrahimy
Second Reader: Hans Bodlaender

July 2025

Contents

Abstract

Heart disease remains one of the leading causes of death worldwide, and accurate early prediction is essential for prevention. In this study, the performance of a custom-built Artificial Neural Network (ANN) is compared to the traditional machine learning models Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) in predicting heart disease from clinical data. Additionally, we investigated feature importance across the models to find which features contribute most to accurate prediction. We implemented the two-layer ANN from scratch using basic Python libraries and compared it to the RF and XGBoost models, which were implemented using built-in libraries. All models were evaluated using 15-fold cross-validation on six performance metrics: accuracy, precision, recall, F1-score, ROC-AUC and log loss. The feature importance was analyzed through the use of built-in functions. We found the ANN outperforming the other models in terms of accuracy, ROC-AUC, and log loss, which suggests that it is more accurate, has better class separation, and is more confident in its predictions. Feature analysis revealed that upward and flat ST-segment slopes, asymptomatic chest pain, and exercise-induced ST-segment depression (oldpeak) were the most influential predictors. These findings suggest that even a simple ANN can be an effective and reliable tool for heart disease prediction. Future studies could work on model interpretability, an increase in model and data set complexity, and the inclusion of other clinical features.

Introduction

Early detection is key in modern medicine, allowing interventions to reduce severity and prevent deaths. Artificial Intelligence (AI) now provides helpful methods, with machine learning models that can help with the identification of patterns that may indicate the start of a disease far in advance of symptom severity. An important area in which AI can be applied is cardiovascular medicine. Heart disease is one of the leading causes of hospitalization and deaths worldwide, affecting over 17 million people (World Health Organization, 2021). It refers to the diseases that limit the heart's performance, typically due to constricted or clogged blood vessels that can lead to chest pain, heart attacks, or heart failure.

An effective way to predict heart disease is by training an Artificial Neural Network (ANN) using clinical data and evaluating its performance. ANNs are inspired by the structure and function of the human brain and consist of interconnected nodes, called neurons. Throughout training, the ANN adjusts its internal weights to improve predictions. To find out how good the model is, we can keep track of some metrics that tell us about the accuracy and reliability of its predictions.

These metrics can provide insight into our model, but to actually know how well it performs, we can compare it to other models like Random Forest (RF) and eXtreme Gradient Boosting (XGBoost). They are both decision tree-based methods that combine multiple decision trees (ensemble methods) to increase performance. RF constructs multiple decision trees on random subsets of the features and data, and averages their predictions to produce final predictions. This reduces overfitting[1] and enhances generalization, making it suitable for most tasks. It is less hyperparameter sensitive, thus easier to implement in many scenarios. Whereas XGBoost constructs trees individually, and each next tree attempts to correct errors made by previous trees for achieving the highest model accuracy in general. XGBoost is generally more accurate than RF but needs careful parameter tuning (Desdhanty & Rustam, 2021).

Furthermore, ANN, RF, and XGBoost models have demonstrated impressive performance in clinical diagnosis tasks. Al-Sharif & Abu-Naser (2023) demonstrated that optimized ANN models trained on structured clinical data can achieve high accuracy (92%) in heart disease prediction. Other

[1] *Overfitting* happens when a model learns the training data too well, including noise and outliers, resulting in poor generalization to new data.

studies have highlighted the strengths of RF and XGBoost. For instance, Teja & Rayalu (2025) evaluated a large set of machine learning models on a heart disease dataset and found RF and XGBoost consistently outperforming most of the other models, including the neural network model. However, Subhadra & Vikas (2019) showed that a simple 2-layer neural network is able to outperform RF, XGBoost, and other models.

One common weakness of previous research is the tendency to place great emphasis on optimized models without properly addressing their generalizability to new clinical contexts or their ability to adapt across different sets of data. Spencer et al. (2020) explored this issue by evaluating a variety of classification models on combined heart disease datasets and found that performance was overly sensitive to the feature selection method used. Their work showed that while some optimized combinations, such as Chi-squared selection with BayesNet, produced strong results, simpler combinations often performed comparably. This suggests that even with highly optimized feature selection and model tuning, better performance is not necessarily guaranteed. While optimization can yield strong results, its benefits may be highly dependent on the context, and simpler combinations can sometimes perform just as well, emphasizing the importance of generalizability over aggressive fine-tuning.

Furthermore, although many studies include feature importance as a secondary analysis, Spencer et al. (2020) made it a central focus, highlighting which clinical attributes (e.g., chest pain, cholesterol levels, and thalassemia) most significantly impact model accuracy. We found that few studies directly compare custom-built neural networks to traditional tree-based models, a gap that limits our understanding of when complex architectures are truly needed.

This thesis aims to bridge this gap by providing an answer to the following research question: *How does the performance of a self-made Artificial Neural Network compare to traditional machine learning models such as Random Forest and XGBoost in predicting heart disease diagnosis based on clinical features?* Additionally, we also aim to answer the following subquestion: *Which clinical features contribute most to accurate heart disease prediction across different models?* We hypothesize that our ANN will perform worse than RF and XGBoost because it is limited by its shallow depth and simplicity (2 layers, no optimization techniques). The other models are optimized and fine-tuned and our ANN has

a simpeler architecture, which might limit its performance. With a more complex architecture and optimization techniques, however, we suspect the ANN could outperform them. We further hypothesize features like chest pain type, maximum heart rate achieved, serum cholesterol, oldpeak, sex, and age contribute most to accurate heart disease prediction, as Talin et al. (2022) showed.

To investigate this, we will build a two-layer ANN from scratch (including forward and backward propagation) and train it on a heart disease dataset. This includes a combination of demographic information, exercise stress test outcomes, chest pain characteristics, blood pressure measurements, blood biomarkers, and ECG results commonly used in cardiovascular diagnostics. After preprocessing and feature selection, a reduced set of the most relevant features will be used for model training. We will then measure its performance on a test set and compare it to the RF and XGBoost models performance using the following evaluation metrics: accuracy, precision, recall, F1-score, ROC-AUC, and log loss. Additionally, we will use feature importance techniques to assess which features most influence the predictions of the models. This not only improves interpretability but also helps to improve the feature set and gives an overview of what features are important.

Methods

For this research, we designed an Artificial Neural Network (ANN) from scratch using the basic Python libraries `NumPy` and `Pandas`, rather than employing high-level machine learning libraries such as `TensorFlow` or `PyTorch`. We visualized the performance using the `matplotlib` and `seaborn` libraries. For RF and XGBoost, we made use of the `scikit-learn` Python library to train and test models on the same dataset.

Dataset

The models were trained on the Heart Failure Prediction Dataset (*Heart Failure Prediction Dataset*, 2021) from Kaggle[2], a merged dataset of five popular heart disease datasets.

Features

The dataset consists of 918 unique observations and includes 11 common clinical features, including numerical, categorical, and binary measurements. The target variable is a binary indicator of heart disease (1 = disease present, 0 = no disease). The dataset had a nearly balanced target distribution,

[2]https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

with 55% positive cases (class 1) and 45% negative cases (class 0). This near balance is unlikely to bias model training or evaluation.

An overview of all features, their units and a brief description of each feature are shown in Table 1. The features that need more clarification will be discussed further.

Table 1: A detailed overview of the features used in the dataset, including their descriptions, data types, and possible values.

Feature name	Description	Type	Values
Age	Age	Numerical	28–77 years
Sex	Gender	Binary	M: Male F: Female
ChestPainType	Type of chest pain experienced by the patient.	Categorical	TA: Typical Angina ATA: Atypical Angina NAP: Non-Anginal Pain ASY: Asymptomatic
RestingBP	Resting blood pressure	Numerical	80–200 mm Hg
Cholesterol	Serum cholesterol level	Numerical	85–603 mg/dL
FastingBS	Blood sugar level while fasting	Binary	1: if FastingBS > 120 mg/dL 0: otherwise
RestingECG	ST abnormalities such as T wave inversions, ST elevation/depression (>0.05 mV), or LVH (left ventricular hypertrophy)	Categorical	Normal ST LVH
MaxHR	Maximum heart rate achieved	Numerical	60–202
ExerciseAngina	Exercise-induced chest pain	Binary	Y: Yes N: No
Oldpeak	ST segment depression during exercise	Numerical	0–6.2
ST_Slope	Slope of the peak exercise ST segment	Categorical	Up: Upsloping Flat: Flat Down: Downsloping

`ChestPainType`

An important indicator of heart disease is chest pain, which can take many different forms, including typical, atypical, non-anginal and asymptomatic chest pain. Typical chest pain often comes as pressure, tightness, or squeezing in the chest. It usually happens when you push yourself physically or

raise your heart rate significantly. Sweating, shortness of breath, or nausea may come with this type of pain, which might radiate to the jaw, neck, or arms. In most cases, it lasts for at least two minutes and gets worse with exercise. On the other hand, atypical chest pain is more likely to feel piercing or sharp. It often does not get better with rest, may vary depending on body posture, and is not caused by physical activity. In most cases, atypical chest pain does not radiate and only lasts a few seconds before quickly going away. Both of these types occur during exercise; when it occurs outside exercise, it is called non-anginal pain. When no chest pain symptoms are experienced, it is called asymptomatic chest pain (Cleveland Clinic, 2023).

RestingECG

The ST segment[3] and T wave on the ECG are the two most critical areas to monitor when interpreting an ECG since they show the repolarization of the heart's electrical cycle. Severe signs of heart disease can be segmental abnormalities, such as ST elevation, T wave inversions, or ST depression[4] (de Alencar et al., 2024). ST elevation is the term used when the ST segment rises above its normal baseline. This is usually seen with an acute myocardial infarction, or heart attack, in which a coronary artery occlusion damages a portion of the heart muscle. It is an important marker that indicates that if the blood supply is not recovered very quickly, the heart tissue will start to degenerate. Whereas, T wave inversion occurs when the T wave, which should be pointing upwards in most ECG outcomes, is reversed. It could indicate reduced blood flow (myocardial ischemia), previous infarct, or other cardiac stress, especially when accompanied by symptoms of chest pain or appearing in multiple results. Left Ventricular Hypertrophy (LVH) is another ECG abnormality, which is thickening of the left ventricle (the major pump chamber of the heart). It is generally a response to long-standing high blood pressure or valvular disease, in which the heart needs to pump harder to pump the blood. Both ST-T abnormalities and LVH are important markers and are used in assessing the risk of heart disease (Deshpande, 2014).

---

[3] The *ST segment* is the flat section between the S wave's end and the beginning of the T wave. It is the interval from ventricular depolarization to the beginning of repolarization (recovery phase). Clinically, depression or elevation of the ST segment may signify myocardial infarct or ischemia (reduced blood flow).

[4] *Depression* here means the amount by which the ST segment drops below baseline, a clinical clue that the heart may not be getting enough oxygen during exertion.

ST_Slope

The ST_Slope feature refers to the slope of the ST segment's peak when exercising, which can be upsloping, flat, or downsloping. ST slope is also a well known marker for myocardial ischemia (Hänninen et al., 2001).

Oldpeak

The Oldpeak feature in a cardiac context means the depression observed between the first and second part of the ST segment in an ECG. It is a comparison between ST-segment depression during exercise and during rest. It is a measurement that indicates reduced blood flow to the heart muscle (myocardial ischemia) (Lin et al., 2023).

Preprocessing

Through analyzing the data, we found no empty or duplicated values. However, we found thirteen instances that had an Oldpeak value less than zero and one instance that had a restingBP value of zero. Since these values are physiologically implausible, we removed these fourteen instances from the dataset. Another feature that had implausible values was Cholesterol, where 172 instances had a cholesterol level of zero. We treated these instances differently, because they represent a substantial portion of the data, by replacing the zero value with the median of the cholesterol values.

Categorical features Sex, ExerciseAngina, ChestPainType, RestingECG and ST_Slope were one-hot encoded in order to use them in the models, i.e., every category has a binary feature corresponding to it. This resulted in a total amount of 20 features. Numerical features such as Age, Cholesterol, and Oldpeak were normalized using z-score normalization (StandardScaler from sklearn), ensuring that each feature has a mean of 0 and a standard deviation of 1. This allows all features to contribute equally to the learning process.

Once the features had been cleaned and encoded, the dataset was transformed into a NumPy array and then shuffled. Data was transposed so that each column had a training example. Training and test sets were established with an 80/20 split, and both were reshaped based on the dimensional requirements of matrix operations in forward and backward propagation.

Feature Selection

To make our models as efficient as possible, we selected the following features as the final dataset: ChestPainType_ASY, ST_Slope_Up, ST_Slope_Flat, Oldpeak, Sex_F, FastingBS, MaxHR,

`ExerciseAngina_N`, `Age`, and `Cholesterol`. These were selected by using the built-in feature importance functions from the RF and XGBoost models and by looking at the permutation feature importance results from every model. From this we found the following features to be the least useful for our models: `ChestPainType_NAP`, `ChestPainType_ATA`, `ChestPainType_TA`, `ST_Slope_Down`, `RestingECG_ST`, `RestingECG_Normal`, `RestingECG_LVH`, and `RestingBP`.

`ExerciseAngina_Y` and `Sex_M` were excluded due to their complementary relationship with `ExerciseAngina_N` and `Sex_F`, which already produced the same binary distinction. We dropped these features to improve model performance and to reduce noise and potential redundancy in the input data, making sure to enhance generalization and model interpretability.

Different Models

Neural Network

Our ANN was made with a simple two-layer architecture. The 10 features used in the final dataset are represented by the 10 units in the input layer $a^{[0]}$. The 10 features were eliminated based on feature importance analysis from an initial collection of 20 variables, which included five numerical, six categorical, and nine one-hot encoded dummy variables.

The hidden layer $a^{[1]}$ contains 10 units and uses the `ReLU`(Rectified Linear Unit) activation function. We used `ReLU` as activation function because of its popularity and the fact that it allows models to learn faster and perform better. The number of neurons was chosen in such a way that they provide a balance between computational simplicity and the performance of the model so that the network remains easy enough to learn useful patterns from the input without any redundancy. Fewer neurons than this can result in underfitting[5], and much higher numbers can result in overfitting given the dataset size limitations.

The output layer $a^{[2]}$ is a single neuron with `sigmoid` activation that generates a probability value between 0 and 1 and is suitable for binary classification tasks such as heart disease presence or absence prediction. The shape of every variable in the architecture is shown in Appendix Figure 3, and the activation functions used are summarized in Appendix Figure 4.

[5] *Underfitting* occurs when a model is overly simplistic in capturing the underlying patterns in the data, resulting in poor performance on both training and testing sets.

Forward Propagation

We initialized the model parameters with small random values and trained the model using gradient descent. In each iteration, we performed forward propagation, in which the input feature vector X is multiplied by a weight matrix $W^{[1]}$, added to a bias $b^{[1]}$, and passed through the `ReLU` activation:

$$\boldsymbol{Z^{[1]} = W^{[1]} X + b^{[1]}} \quad \longrightarrow \quad \boldsymbol{A^{[1]} = \mathbf{ReLU}\left(Z^{[1]}\right)}$$

The resulting intermediate value $Z^{[1]}$ represents the linear combination of inputs before activation. The hidden layer output $a^{[1]}$ is then passed to the output layer:

$$\boldsymbol{Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}} \quad \longrightarrow \quad \boldsymbol{\hat{y} = \mathbf{sigmoid}\left(Z^{[2]}\right)}$$

Backward Propagation

The predicted output $\hat{y}$ is then compared to the true label Y to estimate the error of the prediction. Backpropagation then produces gradients dZ, dW and db as follows, where m is the number of instances and T indicates a transpose operation:

$$\boldsymbol{dZ^{[2]} = Y - \hat{y}} \quad \longrightarrow \quad \boldsymbol{dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}} \textbf{ and } \boldsymbol{db^{[2]} = \frac{1}{m} \sum dZ^{[2]}}$$

Next, the error is propagated to the hidden layer, where g' is the derivative of `ReLU` (1 if Z > 0, else 0). Using this, the gradients for the first layer can be computed as follows:

$$\boldsymbol{dZ^{[1]} = W^{[2]T} dZ^{[2]} \cdot g'\left(Z^{[1]}\right)} \quad \longrightarrow \quad \boldsymbol{dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]T}} \textbf{ and } \boldsymbol{db^{[1]} = \frac{1}{m} \sum dZ^{[1]}}$$

These gradients are then used to iteratively update the weights and biases in the direction that minimizes the loss. The complete set of equations is shown in Appendix Figure 2.

Random Forest

For the RF model, we used the `RandomForestClassifier` class from the `scikit-learn` library (version 1.6.1). The model was set to 1,000 decision trees (`n_estimators`=1000) so that the prediction would be consistent and the variation low. We set the maximum depth of each tree to 4 (`max_depth`=4) to avoid overfitting and ensure the model captured general patterns rather than noise in the training data. Additionally, we used `log_loss` as the splitting criterion, which is a measure that assesses the purity of splits based on expected probability distributions rather than class labels. This choice promotes the

model to produce well-calibrated probability predictions, which is especially important in clinical settings where risk scores, rather than strict classifications, inform decision-making.

XGBoost

The XGBoost model was implemented with the `XGBClassifier` from the XGBoost package (version 3.0.2). Given the sensitivity of boosting models to parameter choices, we used a randomized search with 200 parameter combinations to optimize important hyperparameters such as learning rate, maximum tree depth, number of boosting rounds, and subsample ratio. This randomized search was done using 5-fold cross-validation to ensure that performance estimates were consistent across different subsets of the training dataset. We again chose `log_loss` as the scoring function, with a focus on parameters that produced confidence and well-calibrated probability results.

Evaluation Metrics

The metrics used for evaluation were Accuracy, Precision, Recall, F1-Score, ROC-AUC and Log Loss. These metrics use four essential components to evaluate the classification model's performance:

- True Positives (TP): Cases where the model correctly predicted heart disease.
- True Negatives (TN): Cases where the model correctly predicted no disease.
- False Positives (FP): Cases where the model incorrectly predicted heart disease when it was actually no disease.
- False Negatives (FN): Cases where the model incorrectly predicted no disease when it was actually heart disease.

Accuracy is an important metric for evaluating the performance of a model. It tells us the ratio of instances correctly classified as heart disease or no disease. However, accuracy can be misleading, especially when dealing with imbalanced datasets where correctly identifying important minority classes is critical. A model may achieve high overall accuracy by primarily predicting the majority class while still performing poorly on the minority class, which is often the class of greatest interest. It is calculated as:

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{Total predictions}}$$

Precision is the ratio of correct positive predictions of the model (heart disease), and recall is the ratio of true positive cases correctly predicted by the model.

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}} \qquad \textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}}$$

Together, they tell us about how extensive and accurate the model is in selecting the positive class. The F1-score is a balanced measure of precision and recall since it is the harmonic mean between the two. A model that performs more precisely and robustly is indicated by a higher F1-score.

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

Area Under the Curve (AUC) is one of the most used metrics for binary classification models such as heart disease prediction. It represents the area under the Receiver Operating Characteristic (ROC) curve, which is a graphical plot of the True Positive Rate (recall) against the False Positive Rate ($\frac{\text{FP}}{\text{FP} + \text{TN}}$) at various thresholds. The ROC-AUC score reflects how well the model differentiates between classes, in our case correctly classifying whether an individual does or does not have heart disease. The greater the ROC-AUC score, the more accurately the model differentiates between these instances at any classification threshold. An ROC-AUC of 0.5 indicates that the model is as good as random guessing, while a perfect model achieves a score of 1.0. This metric is particularly handy in clinical applications since it evaluates model performance independent of the classification threshold, which is preferable when the cost of false positives (e.g., unnecessary tests) and false negatives (e.g., not diagnosing heart disease) must be carefully balanced.

Log loss, also known as logarithmic loss or cross-entropy loss, is a measure for determining the accuracy of a classification model based on the probabilities associated with each prediction. It not only determines whether a prediction was correct or not, but also how confident the model was in making that prediction. Whereas metrics such as accuracy or F1-score treat all correct predictions equally well, log loss penalizes overconfidence in wrong predictions more. For example, if the model predicted that a patient has heart disease with 99% probability and that was incorrect, the log loss will be much higher than when the model predicted a more cautious estimate of 60% chance and was also incorrect. This makes log loss particularly useful in contexts like clinical diagnosis, where misclassifying a patient with high confidence is potentially harmful and misleading. Log loss is also more informative than accuracy when we care about the quality of the predicted probabilities. For instance, a model might be 95%

accurate by always predicting the majority class, but if it does so with high confidence even for false predictions, it might still have high log loss, indicating the miscalibration and overconfidence of the model. It is calculated as follows:

$$\mathbf{LogLoss} = -\frac{\mathbf{1}}{\mathbf{m}}\sum_{\mathbf{i=1}}^{\mathbf{m}}[y_{\mathbf{i}}\log(\hat{y}_{\mathbf{i}}) + (\mathbf{1} - y_{\mathbf{i}})\log(\mathbf{1} - \hat{y}_{\mathbf{i}})]$$

where,
$m =$ number of instances
$y_i =$ true label of the i^{th} sample
$\hat{y}_i =$ predicted probability of class 1 for the i^{th} sample

For our heart disease prediction problem, recall is one of the most important metrics, because classifying a patient with heart disease as healthy can be extremely dangerous. It is critical to not miss a sick patient. F1–score is also important, as we are working with a slightly imbalanced dataset and want a balanced performance indicator. Furthermore, log loss tells us how confident our model is and is useful when assigning a risk score (e.g., "90% chance of disease"). ROC-AUC plays a critical role in determining which model would most effectively separate diseased from healthy patients and is therefore also a very important metric.

Data Analysis Procedures

We trained the ANN with a learning rate[6] of 0.003 and adjusted the parameters over 50,000 iterations to ensure convergence. The learning rate was selected after some preliminary experimentation, in which higher values (e.g., 0.01) resulted in unstable training and overfitting, while lower values (e.g., 0.001) caused significant underfitting of the model. A value of 0.003 gave a good balance between convergence rate and stability, giving the optimal solutions in terms of loss minimization and evaluation metrics. The choice of 50,000 iterations was so that the model would have plenty of time to converge without underfitting or ending too soon. Training beyond this point resulted in very small changes and the risk of overfitting.

The models were then evaluated on the test set as well as on the training set to make sure there was little to no overfitting taking place. This is important to make sure the model generalizes well and thus works on different subsets of the data.

[6] *Learning rate* is a hyperparameter that controls the magnitude of steps a machine learning algorithm will make during weight updating when training; e.g., learning rate 0.01 means the model adjusts its weights by 1% of the calculated gradient during each training step.

After training and evaluating each model, we ran a 15-fold cross-validation algorithm on each model, resulting in a set of scores for every metric for every model. We used `StratifiedKFold` to ensure each fold has the same proportion of classes, with 15 splits to have enough scores to make reliable tests. This is done to preserve class proportions across folds to get more reliable validation.

To compare the performance of the ANN to the RF and XGBoost models, we used the Wilcoxon signed-rank test at the significance threshold $p < 0.05$. It is a non-parametric statistical test to compare two related samples. It is a good alternative to the paired t-test when data does not pass the normality assumption required for the t-test or when we cannot assume normality. In our case, we did not check for normality specifically, and we therefore chose the Wilcoxon test, which does not require the assumption of normality. The test ranks paired observations by the size of their absolute values and takes their sign (positive / negative) into account. For our case, the paired samples were the model performance measures (e.g., accuracy, ROC-AUC) for the same cross-validation folds. The W-value in the Wilcoxon signed-rank test is obtained by ranking absolute differences between paired observations (discarding zeros), assigning a positive or negative sign to each rank based on whether the first observation is larger or smaller than the second, and summing up these signed ranks. The null hypothesis (H_0) of this test is that there is no consistent difference in performance between both models. A statistically significant result ($p < 0.05$) leads to rejection of the null hypothesis, with one model being consistently better on all folds than the other.

We also measured feature importance using the `permutation_importance` from `scikit-learn`. Permutation importance was used to estimate the impact of each feature on model performance by measuring the decrease in a chosen metric when the feature's values are randomly shuffled. We used `neg_log_loss` as the scoring for feature importances.

All the code used to implement and evaluate the models is publicly available on GitHub[7].

[7]https://github.com/andreas4589/heart-disease-prediction

Results

The mean values of the evaluation metrics obtained by 15-fold cross-validation are presented in Table 2. Results of the Wilcoxon signed-rank test, as well as the significance of observed differences, are presented in Table 3.

No significant differences in recall, precision, or F1-score between ANN, RF, and XGBoost were observed ($p > 0.05$), indicating similar model performance on these metrics.

The ANN model achieved the highest accuracy at 88.4%, which was significantly higher than both the RF model (86.0%, $W = 8.5$, $p = 0.0165$) and the XGBoost model (86.8%, $W = 5$, $p = 0.0371$). models. The ANN also got the best ROC-AUC score of 94.0%, significantly outperforming RF (93.0%, $W = 11.5$, $p = 0.0043$) and XGBoost (92.8%, $W = 7$, $p = 0.0012$), reflecting better class separation. Additionally, the ANN demonstrated greater confidence in its predictions, as shown by its significantly lower log loss of 30.9%, compared to RF (34.1%, $W = 13$, $p = 0.0054$) and XGBoost (33.5%, $W = 12$, $p = 0.0043$).

Figure 1 shows the permutation feature importance for each model. An example of a true negative and a true positive prediction of the ANN is given in appendix Table 4 and Table 5. This could give extra insight into how confident the model is in its predictions, and what values of the features resulted in the predictions.

Table 2: Mean ± standard deviation of evaluation metrics over 15-fold cross-validation. All models were evaluated on the same test splits.

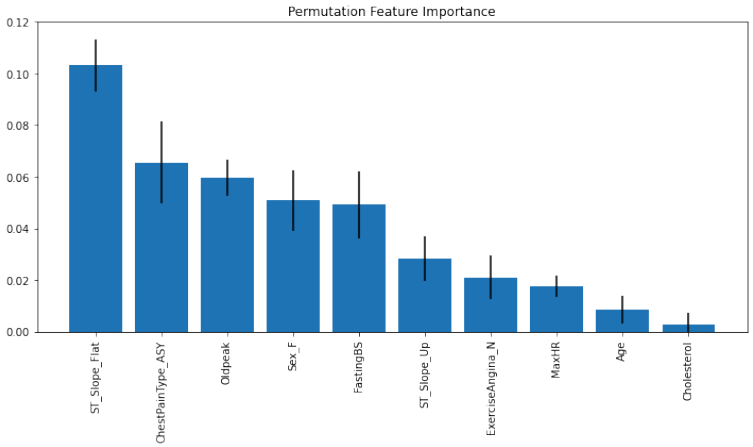
Metric	**ANN**	**RF**	**XGBoost**
Recall	0.9055 ± 0.0654	0.9036 ± 0.0495	0.8996 ± 0.0548
F1–score	0.8906 ± 0.0373	0.8762 ± 0.0289	0.8825 ± 0.0340
Log loss	0.3091 ± 0.0722	0.3406 ± 0.0619	0.3354 ± 0.0733
ROC–AUC	0.9395 ± 0.0386	0.9301 ± 0.0371	0.9281 ± 0.0366
Accuracy	0.8838 ± 0.0393	0.8595 ± 0.0321	0.8683 ± 0.0378
Precision	0.8719 ± 0.0442	0.8534 ± 0.0450	0.8693 ± 0.0462

Table 3: Results of the Wilcoxon signed-rank tests comparing model performance across metrics. For each metric, the table indicates whether the differences between model pairs were statistically significant ($p < 0.05$). Significant results suggest that one model consistently outperforms the other on that metric.

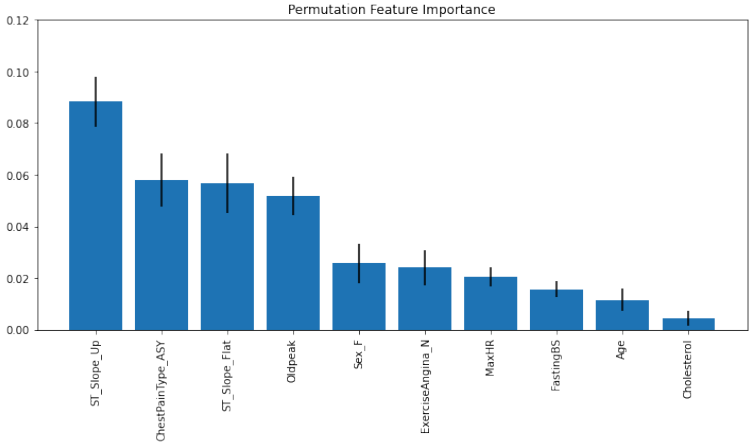
Comparison	**Metric**	**W–value**	**p–value**	**Result**
ANN vs RF	Recall	19.5000	0.7206	Not significant
ANN vs XGBoost	Recall	37.5000	0.5712	Not significant
ANN vs RF	F1–score	18.0000	0.0994	Not significant
ANN vs XGBoost	F1–score	20.0000	0.1361	Not significant
ANN vs RF	Log Loss	13.0000	0.0054	Significant ($p < 0.05$)
ANN vs XGBoost	Log Loss	12.0000	0.0043	Significant ($p < 0.05$)
ANN vs RF	ROC AUC	11.5000	0.0043	Significant ($p < 0.05$)
ANN vs XGBoost	ROC AUC	7.0000	0.0012	Significant ($p < 0.05$)
ANN vs RF	Accuracy	8.5000	0.0165	Significant ($p < 0.05$)
ANN vs XGBoost	Accuracy	5.0000	0.0371	Significant ($p < 0.05$)
ANN vs RF	Precision	22.0000	0.1005	Not significant
ANN vs XGBoost	Precision	35.0000	0.7537	Not significant

Figure 1: Permutation feature importance of input variables used in the models. Subfigures show results for (a) ANN, (b) RF, and (c) XGBoost. The importance scores on the y-axis reflect the average decrease in model performance (e.g., accuracy or F1) when each feature's values are randomly shuffled. Higher bars indicate greater impact on model predictions. Error bars represent standard deviations across multiple permutations.

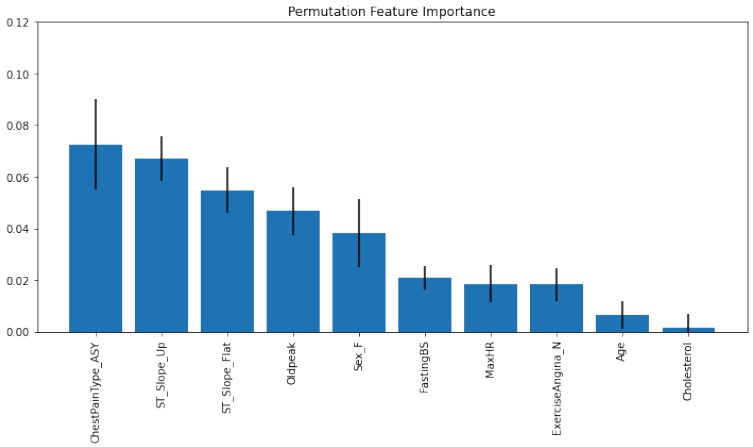
(a) ANN



(b) RF



(c) XGBoost



Discussion

The aim of this research was to investigate how a simple ANN compares to traditional machine learning methods for heart disease prediction. The strongest finding, as can be seen in Table 2 and Table 3, was that the ANN outperformed the other models on accuracy, ROC-AUC and log loss and performed similarly on the other metrics. While the differences were relatively small, the ANN still had a significantly higher accuracy (88.4% vs. 86.0% for RF and 86.8% for XGBoost), showing a higher rate of correct overall classifications. More importantly, its better ROC–AUC (94.0% vs. 93.0% for RF and 92.8% for XGBoost) indicates that it is more capable of distinguishing between heart disease and non-disease cases on different thresholds. The lower log loss of the ANN (30.9% vs. 34.1% for RF and 33.5% for XGBoost) indicates that it is better at giving well-calibrated probabilities and is less overconfident in making incorrect predictions. This is important to give clinicians reliable confidence estimates when making diagnostic decisions. Furthermore, the ANN (as well as the other models) achieved impressive recall, indicating minimal false negatives. This is especially important in this context, since false negatives could result in undiagnosed heart disease with potentially fatal consequences.

Contrary to our hypothesis, the ANN's performance was unexpected. We expected that the more complex tree-based models, RF and XGBoost, would outperform the simpler two-layer ANN, as the majority of research verifies the benefit of ensemble techniques over neural networks. Our results challenge this assumption with the notion that even a simple neural network can perform exceptionally well in heart disease prediction if implemented properly. While tree-based ensemble models will always be a popular choice for their performance and interpretability, our results indicate that model complexity does not always lead to improved predictive capability. Supporting this, Subhadra & Vikas (2019) showed that a simple two-layer neural network is capable of beating more traditional models in this context, particularly when fine-tuned and trained on well-selected features. All this indicates that low-complexity neural architectures should be used more in clinical prediction tasks, particularly when effectiveness and simplicity of deployment are crucial.

Besides comparing overall model performance, we investigated which clinical features contribute most to accurate heart disease prediction across different models. As shown in Figure 1, `ST_Slope_Up`, `ST_Slope_Flat`, `ChestPain_ASY`, and `Oldpeak` ranked highly across all models. This finding aligns with clinical knowledge, as ST segment changes during exercise testing are established indicators of coronary artery disease (a heart disease type) (Shahjehan et al., 2024), while asymptomatic chest pain is a known characteristic of heart disease patients (Liu-Fei et al., 2023). `Sex_F` also ranked highly, which indicates that sex-based physiological differences provide significant predictive value. These results partially confirm our hypothesis and align with previous research. As Talin et al. (2022) showed, features such as `ChestPainType` and `Oldpeak` are important indicators of heart disease. However, our findings differ from some previous research regarding ST segment patterns. While `ST_Slope_Up` and `ST_Slope_Flat`, were among the most important features in all of our models, a down-sloping ST segment was a more accurate predictor in previous research (Stern, 2002). One explanation that `ST_Slope_Down` has low feature importance throughout our models may be that `ST_Slope_Down` is highly correlated with the `Oldpeak` feature. Since both are indicators of exercise-induced ischemia, `Oldpeak` may substitute the predictive ability of `ST_Slope_Down` as redundant during model training. Surprisingly, traditionally important clinical markers showed lower predictive power. `Age` and `Cholesterol` were the least important variables in all models, despite their regular usage in heart disease models. This could be because they share their effect with other features. The cholesterol feature may have been biased due to the fact that zero values were replaced with the median, potentially distorting its natural distribution and weakening its predictive power. Features such as `ExerciseAngina_N`, and `MaxHR`, while traditionally part of clinical assessment, showed only moderate importance for our models. This indicates that within this data set, they are not as strong as additional predictors when there already are stronger predictors such as `ST_slope` and `ChestPainType`.

The clinical relevance goes beyond computational efficiency and reveals an interesting finding: even a simple custom-built ANN achieves performance that is competitive with or superior to more complex approaches like RF and XGBoost. With 88.4% accuracy, 30.9% log loss, and 94.0% ROC-AUC, our ANN demonstrates that simple architectures can lead to clinically suitable performance for heart disease prediction. This suggests that more advanced and optimized neural network designs may

perhaps produce an even better outcome, opening pathways for advanced diagnostic capabilities without sacrificing practical implementation benefits. The high recall achieved across all models is especially valuable in clinical settings, where missing a positive case (false negative) could have life-threatening consequences. Our ANN's ability to minimize overconfident incorrect predictions (low log loss) demonstrates that simpler architectures can maintain the reliability crucial for clinician trust and appropriate treatment decisions. If such a design achieves this level of performance, more advanced neural networks could potentially deliver breakthrough diagnostic accuracy.

While these results give valuable insights into the effectiveness of ANN models for heart disease prediction, several limitations suggest important future research opportunities. Our dataset lacked some clinically relevant features known to influence heart disease prediction. For instance, Talin et al. (2022) and Spencer et al. (2020) found `ca` (number of major blood vessels) and `thal` (thalassemia[8]) to be important predictors. Their absence in our data set could have limited the model performance. Features like these should be included in future research to improve the performance of the models and the generalization. The simplicity of the ANN can be another drawback, since the model only had two layers and we used no regularization techniques or hyperparameter optimization. Further studies can be done on whether more layers are able to further improve its performance. The relatively low sample size of the data set may limit the generalization and increase the risk of overfitting, particularly for more complex models. Our ANN continues to be a black box in decision-making despite its outstanding performance. To make it easier to understand, future studies can include Explainable AI (XAI) techniques. Interpretability is crucial in clinical settings since it would be able to guarantee transparency and give useful insights. With these approaches applied, one would be able to gain more insight into what features are influencing decisions, why particular predictions are being made, and how confident the model is in various areas of the feature space. A final drawback of this research is the fact that the comparison between a custom-built (non-optimized) ANN and optimized models can be unfair, since the ANN was not subjected to the same level of hyperparameter tuning and optimization as the other models.

[8] A genetic blood condition where the body fails to make enough hemoglobin.

This study demonstrates that a custom ANN can be very powerful and competitive with other very popular models. ANN significantly outperformed RF and XGBoost in accuracy (88.4%), ROC-AUC (94%), and log loss (30.9%), challenging assumptions in clinical machine learning. Beyond performance, the results have important, useful consequences. The identification of ST segment features (`ST_Slope_Up`, `ST_Slope_Flat`, `Oldpeak`), and `ChestPain_ASY` as the most predictive features validates clinical knowledge while revealing that traditional indicators like age and cholesterol may be less critical when stronger indicators are present. More importantly, the superior performance of a simple model opens new possibilities for accessible, deployable clinical decision support tools. These results suggest that models prefer feature selection and good model design over architectural complexity. For heart disease cases where ease of implementation, correctly calibrated probabilities, class discrimination, and minimal false negatives are critical issues, carefully designed simple neural networks could offer the optimal tradeoff between accuracy, simplicity, and clinical utility. This work encourages wider availability of efficient diagnostic tools, especially in settings where advanced ensemble models may be impractical but precise heart disease prediction is still vital.

References

Al-Sharif, A. M. H., & Abu-Naser, S. S. (2023). Predicting Heart Disease Using Neural Networks (1st edition). *International Journal of Academic Information Systems Research (IJAISR)*, *7*(9), 40–46.

Cleveland Clinic. (2023, April). *Atypical Chest Pain*. https://my.clevelandclinic.org/health/symptoms/24935-atypical-chest-pain

de Alencar, J. N., de Andrade Matos, V. F., Scheffer, M. K., Felicioni, S. P., De Marchi, M. F. N., & Martínez-Sellés, M. (2024). ST segment and T wave abnormalities: A narrative review. *Journal of Electrocardiology*, *85*, 7–15. https://doi.org/10.1016/j.jelectrocard.2024.05.085

Desdhanty, V. S., & Rustam, Z. (2021). Liver Cancer Classification Using Random Forest and Extreme Gradient Boosting (XGBoost) with Genetic Algorithm as Feature Selection. *2021 International Conference on Decision Aid Sciences and Application (DASA)*, 716–719. https://doi.org/10.1109/DASA53625.2021.9682311

Deshpande, A. (2014). ST-segment elevation: Distinguishing ST elevation myocardial infarction from ST elevation secondary to nonischemic etiologies. *World Journal of Cardiology*, *6*(10), 1067. https://doi.org/10.4330/wjc.v6.i10.1067

Heart Failure Prediction Dataset. (2021, September 10). Kaggle. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

Hänninen, H., Takala, P., Mäkijärvi, M., Korhonen, P., Oikarinen, L., Simelius, K., Nenonen, J., Katila, T., & Toivonen, L. (2001). ST-segment level and slope in exercise-induced myocardial ischemia evaluated with body surface potential mapping. *The American Journal of Cardiology*, *88*(10), 1152–1156. https://doi.org/10.1016/s0002-9149(01)02052-5

Lin, Z., Chen, S., & Chen, J. (2023). Exploring Heart Disease Prediction through Machine Learning Techniques. *Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering*, 964–969. https://doi.org/10.1145/3650400.3650563

Liu-Fei, F., McKinney, J., & McManus, B. M. (2023). Viral Heart Disease: Diagnosis, Management, and Mechanisms. *Canadian Journal of Cardiology*, *39*(6), 829–838. https://doi.org/10.1016/j.cjca.2023.03.020

Shahjehan, R., Sharma, S., & Bhutta, B. (2024). *Coronary Artery Disease*. StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK564304/

Spencer, R., Thabtah, F., Abdelhamid, N., & Thompson, M. (2020). Exploring Feature Selection and Classification Methods for Predicting Heart Disease. *Digital Health*, *6*. https://doi.org/10.1177/2055207620914777

Stern, S. (2002). State of the Art in Stress Testing and Ischaemia Monitoring. *Cardiac Electrophysiology Review*, *6*(3), 204–208. https://doi.org/10.1023/a:1016364622124

Subhadra, K., & Vikas, B. (2019). Neural network based intelligent system for predicting heart disease. *International Journal of Innovative Technology and Exploring Engineering*, *8*(5), 484–487.

Talin, I. A., Abid, M. H., Khan, M. A., Kee, S., & Nahid, A. (2022). Finding the Influential Clinical Traits That Impact on the Diagnosis of Heart Disease Using Statistical and Machine-Learning Techniques. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-24633-4

Teja, M. D., & Rayalu, G. M. (2025). Optimizing heart disease diagnosis with advanced machine learning models: a comparison of predictive performance. *BMC Cardiovascular Disorders*, *25*(1), 212.

World Health Organization. (2021, June 11). *Cardiovascular diseases (CVDs)*. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Appendix

Figure 2: Forward & Backward Propagation

Forward propagation

$$Z^{[1]} = W^{[1]}A^{[0]} + b^{[1]}$$

$$A^{[1]} = g_{\text{ReLU}}\left(Z^{[1]}\right)$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = \hat{y} = g_{\text{sigmoid}}\left(Z^{[2]}\right)$$

Backward propagation

$$dZ^{[2]} = Y - \hat{y}$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$$db^{[2]} = \frac{1}{m} \sum dZ^{[2]}$$

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} \cdot \left(g_{\text{ReLU}}\right)'\left(Z^{[1]}\right)$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]T}$$

$$db^{[1]} = \frac{1}{m} \sum dZ^{[1]}$$

Parameter updates

$$W^{[2]} = W^{[2]} - \alpha dW^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha db^{[2]}$$

$$W^{[1]} = W^{[1]} - \alpha dW^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha db^{[1]}$$

Figure 3: Dimensions of Variables

Variable shapes

Forward propagation:

$A^{[0]} = X$: n × m

$Z^{[1]} \sim A^{[1]}$: 10 × m

$W^{[1]}$: 10 × n (as $W^{[1]}A^{[0]} \sim Z^{[1]}$)

$b^{[1]}$: 10 × 1

$Z^{[2]} \sim A^{[2]}$: 1 × m

$W^{[2]}$: 1 × 10 (as $W^{[2]}A^{[1]} \sim Z^{[2]}$)

$b^{[2]}$: 1 × 1

Backward propagation:

$dZ^{[2]}$: 1 × m ($A^{[2]}$)

$dW^{[2]}$: 1 × 10

$db^{[2]}$: 1 × 1

$dZ^{[1]}$: 10 × m ($A^{[1]}$)

$dW^{[1]}$: 10 × n

$db^{[1]}$: 10 × 1

Figure 4: Activation Formulas

Activation formulas

ReLU : $f(x) = \begin{cases} x \text{ if } x>0 \\ 0 \text{ if } x\leq 0 \end{cases}$

Sigmoid : $\sigma = \frac{1}{1+e^{-x}}$

ReLU′ : $g(x) = \begin{cases} 1 \text{ if } x>0 \\ 0 \text{ if } x\leq 0 \end{cases}$

$A^{[0]} = X$: Input data matrix

$A^{[l]}$: Activation output of layer l

$W^{[l]}$: Weight matrix for layer l

$b^{[l]}$: Bias vector for layer l

$Z^{[l]}$: Linear combination of inputs before activation at layer l

T: Transpose operator

g: Activation function (ReLU and sigmoid)

$dZ^{[l]}$, $dW^{[l]}$, $db^{[l]}$: Gradients used in backpropagation

m: Number of training examples

n: Number of features

Y: True target values

α: Learning rate parameter

$A^{[2]} = \hat{y}$: Final output (predicted probability class 1)

Table 4: True negative example with the actual label, predicted label and probability of the prediction. A prediction close to 1 is a very certain positive prediction and close to 0 is a very confident negative prediction. For every feature it shows what value was used in the prediction.

Predicted Class:	**0**	**No Heart Disease**
Actual Label:	0	No Heart Disease
Probability	0.0393	96.07% chance
Feature	**Value**	**Range/Label**
Age	42.0	28 – 77
Cholesterol	211.0	85 – 603
FastingBS	0	1 / 0
MaxHR	137.0	60 – 202
Oldpeak	0	0 - 6.2
Sex_F	1.0	1 (Female) / 0 (Male)
ExerciseAngina_N	1.0	1 (No) / 0 (Yes)
ChestPainType_ASY	0	1 (ASY) / 0 (other)
ST_Slope_Flat	0	1 (Flat) / 0 (other)
ST_Slope_Up	1.0	1 (Up) / 0 (other)

Table 5: True positive example with the actual label, predicted label and probability of the prediction. A prediction close to 1 is a very certain positive prediction and close to 0 is a very confident negative prediction. For every feature it shows what value was used in the prediction.

Predicted Class:	**1**	**Heart Disease**
Actual Label:	1	Heart Disease
Probability	0.9720	97.20% chance
Feature	**Value**	**Range/Label**
Age	60.0	28 – 77
Cholesterol	258.0	85 – 603
FastingBS	0	1 / 0
MaxHR	141.0	60 – 202
Oldpeak	2.8	0 – 6.2
Sex_F	0	1 (Female) / 0 (Male)
ExerciseAngina_N	0	1 (No) / 0 (Yes)
ChestPainType_ASY	1.0	1 (ASY) / 0 (other)
ST_Slope_Flat	1.0	1 (Flat) / 0 (other)
ST_Slope_Up	0	1 (Up) / 0 (other)