Logistic Regression vs. Artificial Neural Networks for Heart Disease Prediction

Chase Roby
Department of Bioengineering
George Mason University
Fairfax, Virginia
croby@gmu.edu

ABSTRACT

Early detection is vital for improving the outcomes of cardiovascular disease, the leading cause of death across the planet. Artificial neural networks can offer some clarity in this field as they are more flexible with complex biomedical datasets than traditional linear models like logistic regression. This report examined the performance between logistic regression and multiple chosen artificial neural network architectures for predicting heart failure using the 'Heart Failure Prediction Dataset' on Kaggle. Through manual testing and cross-validation to determine the highest performers, four ANN structures ([10], [100], [20 10], [40 20 10]) were selected. The four chosen architectures were systematically modified and tested with a variation of learning rates and epoch settings for further tuning. Following this, the best configuration for each of the ANN models was identified and each of the four top performers was assessed in final comparisons against the standard logistic regression model. Each of these steps was quantified using a combination of metrics including accuracy, precision, recall, and F1 score.

For each of the candidate ANN architectures, the results varied widely with logistic regression staying very competitive. The [40 20 10] ANN model was able to demonstrate the strongest overall results and outperformed the simple LR model on three of the four metrics. However, the LR model remained very close in metrics across the repeated runs for validation. The single-layer models ([10], [100]) were able to achieve higher accuracy, precision, and F1 scores with a slightly lower recall than the LR model. Unexpectedly, the structure [20 10] was outperformed on all four metrics by logistic regression. This indicates that even though neural networks can capture more nonlinear relationships, the dataset one is working with plays a major factor in generalization abilities. The results showcase the necessity of forming proper model architecture, hyperparameter tuning, and understanding dataset properties when using machine learning models for biomedical predictions.

I. Introduction

Cardiovascular disease is still the leading cause of death across the globe. Solving this issue remains a mystery in the world of biomedical research, but some of the novel data analytics methods can provide some crucial progress in this sector. Being able to predict these diseases early and reliably will improve lives and help us understand future preventative strategies. Assessing the risk of heart disease in models using logistic regression can limit the capability to predict with accuracy due to the struggle to adjust to nonlinear relationships. The use of machine learning models, for example Artificial Neural Networks (ANN), could serve as a better substitute to interpret complex biomedical data. Heart disease prediction likely doesn't fit into more narrow linear relationships, so utilizing these updated machine learning methods can provide improved predictive outcomes [1].

I hypothesize that an Artificial Neural Network model will have better predictive performance than logistic regression on heart disease. The Artificial Neural Network (ANN) model will pick up on patient subtleties more effectively and will have

higher accuracy, recall, precision, and F-1 scores to quantify better predictive abilities.

Logistic regression has been used to interpret datasets for several decades, meaning there are many use-cases of these models as predictive tools. Machine learning has been a rapidly expanding field since then for analyzing complex biomedical datasets. An example dating back to 1989 utilized purely logistic regression to predict coronary disease and showcased this as a promising predictor [2]. This work has been built off in the coming years for integrating the rising ML methods. Another study examined the relationship between artificial neural networks and logistic regression within the context of scoring risk for breast cancer. The results of this showed that ANNs were able to achieve higher accuracy and sensitivity in risk estimation compared to regression. Although this assesses breast cancer rather than coronary disease, it shows potential for gaining insight into other sectors [3]. In a more recent study, researchers compared models such as k-nearest neighbor, support vector machine, and random forest to see how well they predicted heart disease. These pointed to random forest being the most accurate and precise, with SVM following closely behind. This study examines some of the other machine learning methods that can be used as predictors in similar contexts and emphasizes the importance of cross-validation for evaluating the real applications of these studies [4].

II. Data

A. Data Source and Predictors

This 'Heart Failure Prediction Dataset' was found on kaggle.com [5]. The dataset contains 918 individual observations combining 5 different heart datasets (Cleveland, Hungarian, Switzerland, Long Beach, and Statlog sets) with 12 common features, which gives a greater sample size to validate the data rather than just 1 of the datasets. The usability score on Kaggle is perfect, meaning it has verifiable credibility and compatibility for handling. The dataset consists of 11 clinical predictors and a final binary prediction of 1 = heart disease, and 0 = no heart disease. The clinical predictors include:

TABLE 1. The predictors included on the Kaggle dataset.

Predictor	**Measurement**	**Type**
Age	Years	Continuous
Sex	Male / Female	Categorical
Chest Pain	TA/ATA/NAP/ASY	Categorical
Resting BP	mmHg	Continuous
Cholesterol	mg/dL	Continuous
Fasting BS	1 if >120mg/dL or 0	Binary
Resting ECG	Normal / ST	Categorical
Max HR	BPM (beats per min)	Continuous
ExerciseAngina	Yes / No	Binary
Oldpeak	ST Depress. Values	Continuous
ST Slope	Up / Flat / Down	Categorical

B. Exploring the Types of Variables

The dataset includes 5 continuous variables; Age, Resting Blood Pressure, Cholesterol, Max Heart Rate, and Oldpeak (examining ST depression). Even though most of the "continuous variables" in this set appear as discrete as they have been rounded to the nearest whole number, all of them could be measured more closely. It also contains 3 binary indicators: Fasting Blood Sugar, Exercise Angina, and the final prediction outcome of heart disease. These are either indicated by Yes/No or 1/0. Finally, it contains 4 strictly categorical variables; Chest Pain Type, Resting ECG, ST Slope, and Sex. Chest Pain Type is measured by classifying as either Typical Angina - TA, Atypical Angina - ATA, Non-Anginal Pain - NAP, or Asymptomatic - ASY. Resting ECG is split into either normal electrocardiogram results or ST wave abnormality (ST or Normal). ST Slope is storing information on the slope of the ST segment taken for those applicable. It is noted in this table as upsloping (Up), flat (Flat), or downsloping (Down). Lastly, Sex is shown as Male or Female.

C. Preprocessing & Missingness

Preprocessing is a challenge when working with this dataset because there are a variety of different data types. One hot encoding will be applied to the categorical variables such as ST_Slope and ChestPainType. This changes the display for all of these and encodes them as a column of 1s and 0s,

making them readable for the machine-learning algorithm. Standardization using z-scores will also be crucial for the continuous variables like cholesterol and age, turning the wide range of values into the number of standard deviations each are away from the mean. No missing values were detected when utilizing the ismissing function within MATLAB, which helps cut down on preliminary work. If any further preprocessing causes changes in missingness, values would be imputed by the median or mode depending on their variable type.

D. Figures

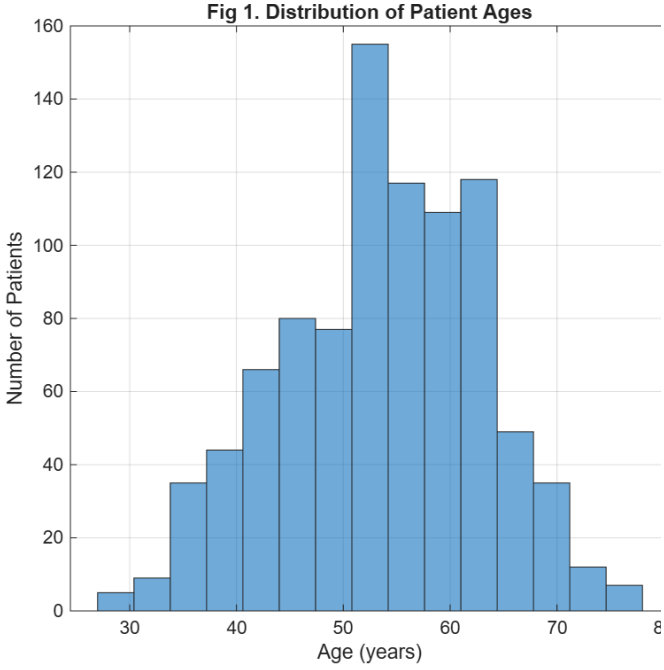

Fig.1. Distribution of Patient Ages. Histogram shows the concentration of patients in the dataset, with a majority being between the ages 45 and 65. Age is one of the more vital and easy to observe predictors and with this distribution being across a wide range it should give more reliable results.

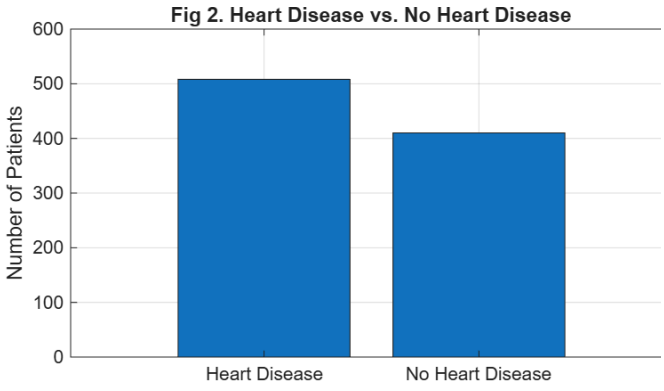

Fig. 2. Heart Disease vs No Heart Disease Class Balance. Bar graph visualizing the balance of the final predictions of heart disease or not for the combined datasets. It results in a relatively close count between the two, meaning no need for a lengthy preprocessing portion to balance them.

III. Methods

A. Logistic Regression

Logistic Regression was chosen as the initial model for means of comparison due to its in-depth usage throughout decades of biomedical research. Logistic regression is a method of classification that utilizes the logistic function on linear relationships of the input variables and outputs a binary prediction of 0 or 1. This method will only fit with linear relationships within predictors, which could hinder its ability compared to newer ML models [6].

B. Artificial Neural Network

Artificial Neural Network (ANN) was selected as the novel machine learning model due to its ability to fit to nonlinear relationships and understand more complex patterns within the data. This will be done utilizing a multi-layer perceptron with a hidden layer or multiple hidden layers. I can use cross-validation to determine the ideal hidden layer size and learning rate.

C. Training, Validation, Testing Sets

Before any model training, the categorical and continuous variables were one-hot encoded and standardized respectively to provide interpretable values for the ML models. The dataset was further partitioned into training, validation, and testing sets. The initial split for the Artificial Neural Network model is 70% for the training set, 15% for the validation set, and 15% for the testing set. This will be done to aggregate 70% to fit the model, 15% to tune the parameters such as hidden layer size and learning rate, and the last 15% for a final evaluation. For the logistic regression model, the same 70% training, 15% validation, and 15% testing will be used to maintain the same partition parameters for each.

D. Model Performance Assessment

The models' performance was assessed using 4 different metrics: accuracy, precision, recall, and F-

1 score. Accuracy is assessing how correct the model is overall. Precision will be addressing the model's ability to classify positive (heart disease) as positive. Recall is examining how many of the positive classifications are correct from those cases that existed. F1-score is balancing the recall and precision scores to more accurately look at their hand-in-hand relationship. Looking at all these statistics together will serve as a solid baseline for assessing how well each of the models perform [7].

E. Context within Hypothesis

The logistic regression and ANN models are appropriate for this problem because they both can model for the binary outcome that is needed in this study. The difference between the two is that logistic regression will only be assuming linear relationships and producing the logarithmic probability of these. The ANN model instead will allow for nonlinear relationships to be utilized which may improve predictive accuracy. By assessing the two models with the same preprocessing and relevant cross-validation techniques, I can analyze whether the extra complexity of the ANN model is better for prediction within this dataset. The knowledge gap within the initial hypothesis will be able to be substantiated using the final accuracy, precision, recall, and F-1 score metrics.

IV. Results

A. Baseline Model Testing

The initial part of this analysis involved understanding the baseline model performance for both the Logistic Regression (LR) and Artificial Neural Network (ANN) models. The simple regression model was established as its final product (with no further changes), while the ANN was intentionally set up with zero hidden layers. This was done to determine if the neural network even with minimal architecture can match the results of the more traditional method.

TABLE 2. Mean performance metrics for baseline models (based on 10 runs for validation).

Model	Accuracy	Precision	Recall	F1
LR	0.8642	0.8724	0.8841	0.8781
ANN	0.8539	0.8659	0.8724	0.8686

As seen with the results in table 2, the logistic regression model was able to outperform the zero hidden layer ANN model in all four of the performance metrics across 10 runs. This was anticipated since there has been no extra configuration that would make the ANN any more capable of capturing complex patterns than logistic regression. Figures 3 and 4 below show an example of one run for each of the baseline models.

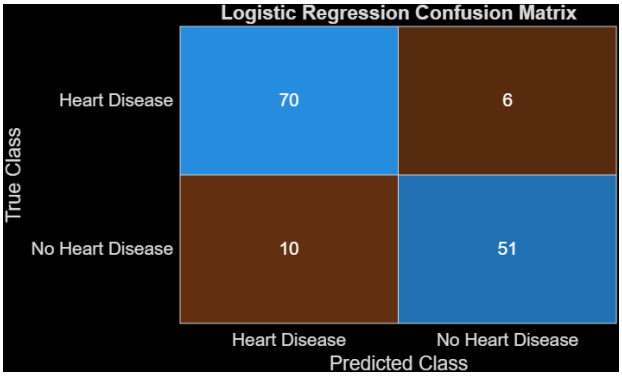

Fig. 3. Confusion Matrix represents one of the runs for the logistic regression model.

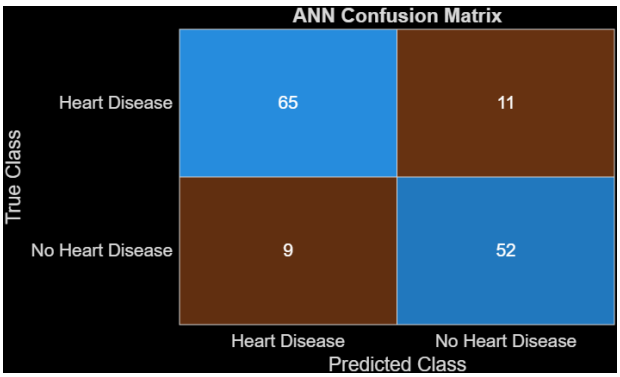

Fig. 4. Confusion Matrix represents one of the runs for the baseline ANN model (no hidden layers).

B. Single-Train Manual Testing

To improve the ANN's performance, the hidden layer was evaluated with two different approaches. First, a simple single-train manual test increasing the complexity of hidden neurons (5, 10, 20, 40, 60, 100, 250, 500, 1000, 10000) was performed to explore the initial ranges of accuracy scores for the single layer. From here, some of the best candidate sizes were chosen for further validation. The results of the single run testing done are shown below in table 3.

TABLE 3. Single run test to gauge accuracy of ANN vs. LR. Determining viable range for 1-layer.

Neuron Size	ANN Accuracy	LR Accuracy
5	0.83942	0.86131
10	0.86131	0.85401
20	0.91971	0.90511
40	0.86861	0.86131
60	0.89051	0.88321
100	0.86131	0.86131
250	0.83212	0.85401
500	0.72263	0.84672
1000	0.82482	0.83942
10000	0.83492	0.86131

C. *Preliminary Findings*

The findings from the quick manual testing show that the hidden layer sizes between 10 and 100 neurons yielded the highest accuracies for the model. These results suggest that even slightly increasing the complexity in the ANN can allow the neural network to outperform or match the performance of the logistic regression model. The ANN model also displayed some impressive performance metrics when the neuron (or node) size was set to 20, with the accuracy being 91.97% and a recall of 96%. These should be taken with skepticism though since these are results of only a single run for each of the neuron sizes, meaning a lot of variation in the randomized sets used when partitioned.

D. *ANN Architecture Cross-Validation*

To further understand and improve the ANN, some of the candidate architectures were examined using a 5-fold cross validation on the training and validation set. I chose five single-layer networks within that 10-100 neuron range as well as four multi-layer additions to see if the added complexity improves the performance. A 5-fold validation was chosen due to the long computing requirements needed when trying to implement a 10-fold. The 5-fold still takes a bit of time to run as well. Although not quite as complex, the 5 k-fold cross validation more comprehensively shows their performances and avoids most of the overfitting to any single validation set. The results from the 5-fold cross validation are shown in Table 4.

TABLE 4. Cross-validation results for each of the architecture types. Rounded to fit in table.

Size	Mean Accuracy	Mean Precision	Mean Recall	Mean F1
[10]	0.8515	0.8464	0.887	0.866
[20]	0.8451	0.8437	0.877	0.859
[40]	0.8425	0.8502	0.861	0.855
[60]	0.8477	0.8654	0.851	0.858
[100]	0.8499	0.8471	0.862	0.853
[20 10]	0.8477	0.8494	0.875	0.861
[40 10]	0.8463	0.8667	0.847	0.856
[60 30]	0.8451	0.8483	0.869	0.857
[40,20,10]	0.8528	0.8513	0.881	0.865

All the different architectural set-ups had relatively high mean accuracies ranging from 84% to 85%. Adding the extra multi-layer networks made it possible to provide some slight improvements in specific metrics like recall and F1 score as well. Varying the depths of the ANN allows you to capture more complex patterns depending on how you precisely tune it. The additional [40, 20, 10] three-layer network was able to achieve the highest overall mean accuracy and very strong metrics like the recall and F-1 score. This implies that adding depth can improve the network in many different facets. Contrary to that, the simple 10-neuron single layer model had some of the best performance when it comes to accuracy and recall. These results indicate that when tuning these parameters like hidden layer size and depth it becomes a delicate process to optimize.

E. *Hyperparameter Testing (Learning Rate/Epochs)*

From here, four of the highest performing architectures were chosen for final comparisons. Chosen were the [10], [100], [20 10], and [40 20 10] architectures. Before setting up the final testing for the updated ANN models, I wanted to manually tune the learning rate and epochs to find the most desirable range. Three different epochs and three learning rates were tested for each of the four ANN architecture. The learning rate was adjusted between 0.001, 0.01, and 0.05 while the epochs were tuned between 150, 300, and 1000. Shown below is a table of the highest performing ANN models per one manual run for each of the learning rate and epoch

settings. The best hyperparameters for each included are on table 5.

TABLE 5. The highest performing run with hyperparameters (LR = learning rate, epochs) tuned for each of the 4 architectures. 3 learning rates and 3 training epoch values tested for each architecture for a total of 9 test runs for each of the four models.

Model	Accuracy	Precision	Recall	F1
ANN [10] LR=0.001, 150 epochs	0.8905	0.8861	0.9211	0.9032
ANN [100] LR=0.01, 300 epochs	0.8832	0.8750	0.9211	0.8974
ANN [20 10] LR = 0.05, 1000 epochs	0.8759	0.8734	0.9079	0.8903
ANN [40 20 10] LR = 0.01, 150 epochs	0.8832	0.8571	0.9474	0.9000

F. Final Comparisons For Logistic Regression vs. Chosen ANN Models

While the highest performances for these different models show the promise they offer, this is not representative of stability or consistency. To solve this, each of the models was tested across 10 repeated runs using the highest performing hyperparameters. The mean metrics (accuracy, precision, recall, and F1 score) were then compared against the performance of logistic regression. Tables 6-9 below present the average metrics for each architecture. With randomized runs, these tables more reliably provide insight on how well they perform.

TABLE 6. Showing the mean performance of the tuned single-layer [10] neural network vs. logistic regression across 10 runs. Architecture [10] set with 0.001 learning rate and 150 epochs.

Model	Mean Acc.	Mean Prec.	Mean Recall	Mean F1
LogReg	0.8504	0.8564	0.8776	0.8663
ANN [10]	0.8511	0.8610	0.8737	0.8663

TABLE 7. Features the mean performance of the tuned single-layer [100] neural network vs. logistic regression across 10 runs. Architecture [100] set with 0.01 learning rate and 300 epochs.

Model	Mean Acc.	Mean Prec.	Mean Recall	Mean F1
LogReg	0.8489	0.8475	0.8882	0.8669
ANN [100]	0.8540	0.8632	0.8789	0.8695

TABLE 8. Presents the mean performance of the tuned multi-layer [20 10] neural network vs. logistic regression across 10 runs. Architecture [20 10] set with 0.05 learning rate and 1000 epochs.

Model	Mean Acc.	Mean Prec.	Mean Recall	Mean F1
LogReg	0.8620	0.8756	0.8763	0.8756
ANN [20 10]	0.8526	0.8630	0.8750	0.8678

TABLE 9. Displays the mean performance of the tuned multi-layer [40 20 10] neural network vs. logistic regression across 10 runs. Architecture [40 20 10] set with 0.01 learning rate and 150 epochs.

Model	Mean Acc.	Mean Prec.	Mean Recall	Mean F1
LogReg	0.8701	0.8818	0.8855	0.8825
ANN [40 20 10]	0.8708	0.8791	0.8908	0.8841

G. Main Findings

Through the four network architectures, the performance metrics over these randomized runs presented some key insights on how model depth and complexity affect classification. The simplest model used, ANN [10], provided very marginal increases in accuracy, precision, and F1 score with a slightly lower recall than logistic regression. Adding slightly more width, the ANN [100] model was able to

improve the average performance even further. The 100-neuron model produced a higher mean accuracy, precision, and F1 score, while logistic regression still held a slight edge in recall.

When digging into the deeper layer architecture, it came with some surprising results. The [20 10] ANN model performed worse in all 4 metrics against logistic regression, indicating it cannot generalize nearly as well. The most complex model, [40 20 10], had overall the highest performing percentages for each of the metrics. The logistic regression model in this comparison performed particularly well also, so the increase between LR and the deepest ANN model was very marginal. The logistic regression model only led the [40 20 10] model in precision by a small degree but was beaten out in the other metrics.

The [40 20 10] model had the most impressive single run across all the final performance testing, achieving an accuracy of 90.51%, a precision of 89.87%, a recall of 93.42%, and F1 score of 91.61%. Confusion matrices of this notably successful run comparing the Logistic Regression and [40 20 10] model are below to visualize the potential improvement that tuning an ANN can make for classifying correctly.

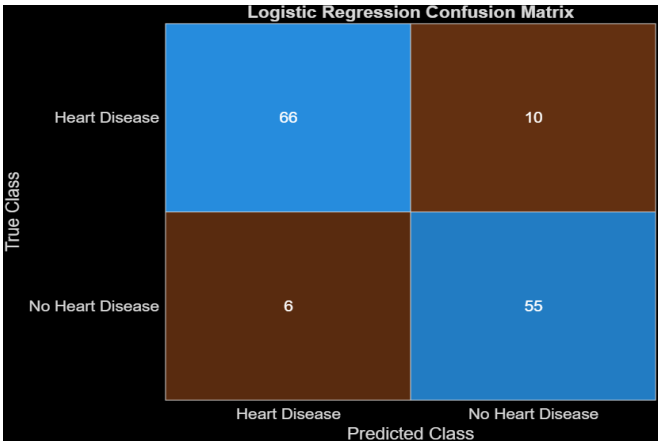

Fig. 5. Logistic Regression Confusion Matrix. Highest individual run for final testing.

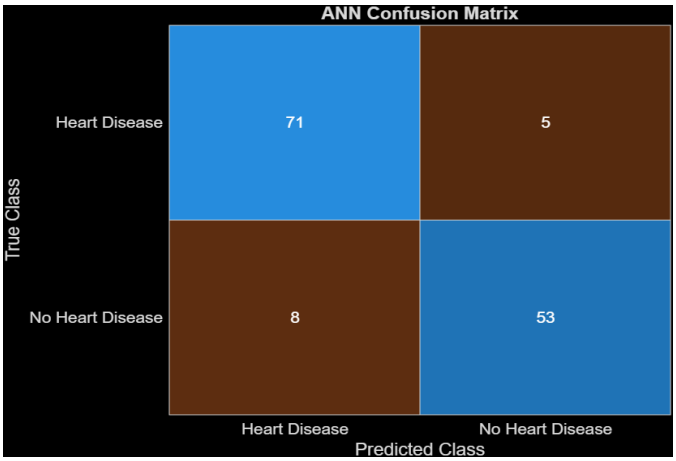

Fig. 6. ANN Confusion Matrix. Highest individual run for final testing. Architecture = [40 20 10].

In total, these results showcase that neural networks of both single-layer and multi-layer can consistently outperform logistic regression in some of the performance metrics.

V. Discussion

This report focused on comparing linear and nonlinear machine learning methods to evaluate how effective each model is for heart disease prediction. Through beginner exploratory analysis, it was shown that the dataset was relatively balanced and had no missing values which suggests that both linear and nonlinear relationships could potentially affect model performance.

Logistic regression was chosen as the baseline model for this study due to its thorough usage in relevant biomedical contexts. Through comparing the initial results from the baseline LR and ANN model, it established a point of reference for further improvement in tuning the neural network. The logistic regression model was able to produce strong accuracy, precision, recall, and F1 scores across validation runs in the baseline testing and outperform ANN across all metrics. These solid numbers signal that the data used is potentially forming some linear relationships, but it could just be performing well due to the smaller/medium tabular dataset utilized.

Through initial manual testing of hidden layer size and subsequent cross-validation techniques, a list of four candidate architectures was selected as “top

performers" given their improved metrics. From this, the four candidate neural network architectures ranging from one-layer to multi-layer were selected to move to further optimization. Each of the four structures ([10], [100], [20 10], [40 20 10]) were subjected to a systematic manual tuning of hyperparameters including learning rate and training epochs, allowing me to distinguish the highest performing configurations for each neural network.

With the optimized learning rate and epoch values for each candidate ANN model, they underwent final comparisons against logistic regression. The mean results across ten randomized runs provided a lot of insight into how these different network architectures behave. The two simple, single-layer ANN models ([10], [100]) were both able to produce higher mean accuracies, precision, and F1 scores than the logistic regression. This demonstrates that even with very little added depth, it increases the network's ability to identify some of the nonlinear structures in the data. The LR model was still able to provide better recall statistics for both single-layer models, however.

On the contrary, the deeper & more complex ANN model [20 10] performed poorly against logistic regression and was trailing in all metrics. This indicates that although there was added depth, there is potentially risk of overfitting due to the dataset's small to moderate size (918 observations). The most complex architecture used, ANN model [40 20 10], provided the highest and most consistent overall performance. It achieved by far the highest accuracy, recall, and F1 score compared to all the other structures, while being just marginally behind logistic regression in precision. This highlights that although some deeper models will have worse metrics, others can still provide more efficient prediction. This is likely to do with extremely precise tuning of hyperparameters to ensure the training is stable and not causing any overfitting. An important note about the [40 20 10] model is the consistently high recall score. High recall will minimize the amount of costly false negatives and is desired in biomedical prediction contexts. Incorrectly labeling a patient of heart disease as healthy carries a ton of risk, so the model producing strong values here is promising.

The study emphasized that neural networks can outperform logistic regression with slight margins of improvement. Although it is dependent on depth, width, and hyperparameter tuning, there are other considerations like dataset size and balance that can cause unlikely results. It also cements the knowledge that simpler models will work better with smaller datasets, since some of the relationships are nonlinear but mildly so. Even with the data-driven and thorough design of the study, the limitations of the dataset size have to be acknowledged and influenced the results produced. A future strategy that could be used to help reduce overfitting and improve generalization is dropout being implemented [8]. Early stopping could also provide some value if researchers can target the ideal range to stop. Simply using a larger or more complex dataset could allow deeper ANN models to reach their full potential as well. Further work into this dataset using other machine learning methods like support vector machines and random forest should be implemented. This would allow one to cross-reference what machine learning method will be best for exactly the dataset size / type that is available.

In conclusion, this project demonstrates that neural network models produce higher peak performances than logistic regression on average. This is only after thorough and complex optimization, however. Logistic regression is still a very stable and robust tool that produces strong predictions in comparison and should remain in use for datasets like this. Even with all the extraneous hidden layer size and hyperparameter tuning, the constraints of the dataset allowed LR to have just marginally worse results. All these findings illustrate the importance of balancing model complexity and dataset size to maximize generalization and further interpretability.

REFERENCES

[1] A. A. Ahmad and H. Polat, "Prediction of Heart Disease Based on Machine Learning Using Jellyfish Optimization Algorithm," *Diagnostics*, vol. 13, no. 14, p. 2392, Jan. 2023, doi: 10.3390/diagnostics13142392.

[2] R. Detrano *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *The American Journal of*

Cardiology, vol. 64, no. 5, pp. 304–310, Aug. 1989, doi: 10.1016/0002-9149(89)90524-9.

[3] T. Ayer, J. Chhatwal, O. Alagoz, C. E. Kahn, R. W. Woods, and E. S. Burnside, "Comparison of Logistic Regression and Artificial Neural Network Models in Breast Cancer Risk Estimation," *Radiographics*, vol. 30, no. 1, pp. 13–22, Jan. 2010, doi: 10.1148/rg.301095057.

[4] Y. Rimal, N. Sharma, S. Paudel, A. Alsadoon, M. P. Koirala, and S. Gill, "Comparative analysis of heart disease prediction using logistic regression, SVM, KNN, and random forest with cross-validation for improved accuracy," *Sci Rep*, vol. 15, no. 1, p. 13444, Apr. 2025, doi: 10.1038/s41598-025-93675-1.

[5] F. Soriano, "Heart Failure Prediction Dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction/data.

[6] S. Sperandei, "Understanding logistic regression analysis," *Biochem Med*, pp. 12–18, 2014, doi: 10.11613/BM.2014.003.

[7] B. Juba and H. S. Le, "Precision-Recall versus Accuracy and the Role of Large Data Sets," *AAAI*, vol. 33, no. 01, pp. 4039–4048, July 2019, doi: 10.1609/aaai.v33i01.3301403

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting".