Predictive Modeling of Coronary Artery Disease Using Neural Networks

Clement Wright

Received September 02, 2024
Accepted December 16, 2024
Electronic access December 31, 2024

Coronary Artery Disease (CAD) is a common condition that affects about 20.5 million adults in the United States, making it the most prevalent type of heart disease in the country, necessitating effective predictive models to aid in early diagnosis and intervention. The study aims to build accurate models capable of predicting the presence of CAD based on various physiological and lifestyle factors. The dataset includes information such as blood pressure, cholesterol levels, chest pain type, resting electrocardiographic results, and exercise-induced angina. The research employs neural network architectures to develop predictive models and evaluates their performance using metrics such as training error, validation error, accuracy, precision, recall, and F1-score metrics. We explain the preprocessing steps, model architecture, and evaluation metrics employed in training and validating the neural network models. Additionally, we discuss the implications of different architectural choices and preprocessing techniques on model performance, including addressing overfitting concerns. Simpler models demonstrated balanced performance with minimal overfitting, while more complex models showed higher accuracy on training data but were overfitted on validation data. The best-performing model achieved an accuracy of 88%, with an even trade-off between precision and recall, reflecting a robust and reliable model that balances sensitivity and specificity, improves upon baseline performance, and holds potential for real-world applications in clinical settings for early intervention.

Keywords: Coronary Artery Disease, Neural Networks, Predictive Modeling, Machine Learning, Early Diagnosis

Introduction

Coronary artery disease (CAD) is responsible for around 610,000 deaths each year in the United States, making it the leading cause of death, representing about one in every four deaths. Globally, CAD ranks as the third leading cause of death, contributing to approximately 17.8 million deaths annually. The financial burden of CAD on the U.S. healthcare system exceeds 200 billion dollars annually. Despite its high prevalence and substantial impact on health and disability, CAD is largely preventable through lifestyle interventions such as smoking cessation, exercise, and dietary changes [1]. However, current diagnostic methods for CAD rely heavily on invasive procedures and clinical judgment, which are not always accessible or cost-effective. For instance, diagnostic tools such as CT angiography, while effective, can be costly and may not be readily available in all healthcare settings, thereby limiting accessibility. These limitations emphasize the pressing need for accurate, non-invasive predictive models to aid in the early diagnosis and intervention of CAD, which are crucial for improving patient outcomes. Predictive modeling using machine learning techniques, particularly neural networks, offers a promising approach for identifying individuals at risk of developing CAD. Neural networks have the capability to capture complex, non-linear relationships within data, making them well-suited for medical diagnosis tasks where interactions between multiple physiological and lifestyle factors are intricate. This paper presents a detailed analysis of building and evaluating neural network models for CAD prediction. We focus mainly on each model's training and validation error, as well as F1-score, accuracy, precision, and recall. Training error measures how well the model fits the training data. It is calculated by comparing the model's predictions against the actual outcomes in the training set. A low training error indicates that the model has effectively learned the patterns within the training data [2–4]. However, a low training error may also signal overfitting, where the model is too closely tuned to the training data and fails to generalize to new data. Validation error, on the other hand, measures the model's performance on a separate validation set, which is not used during the training process. This error indicates how well the model generalizes to new, unseen data. A lower validation error is desirable as it suggests that the model is not only capturing patterns from the training data but also successfully applying those learned patterns to make accurate predictions on new data [2–4].

Literature Review

Recent studies applying machine learning to CAD prediction include a wide variety of models, ranging from simpler statistical methods like logistic regression and decision trees to more advanced approaches such as support vector machines (SVMs), convolutional neural networks (CNNs), and generative adversarial networks (GANs)[5]. For example, Detrano et al. (1989) developed a discriminant function model to estimate probabilities of angiographic coronary disease. Tested across multiple datasets, including the Cleveland dataset, the model demonstrated reliable clinical utility, particularly for patients with intermediate disease prevalence. While the discriminant function outperformed Bayesian approaches like CADENZA in some cases, its clinical utility was modest, stressing the potential need for more sophisticated predictive models that can handle variability across datasets. In more recent years, machine learning approaches have expanded to include decision trees, SVMs, and ensemble-based methods. Bashir et al. (2016)[6] demonstrated the effectiveness of an ensemble model combining decision trees and SVMs, achieving an accuracy of 87.37%. Daraei and Hamidi (2017)[7] applied the J48 algorithm to myocardial infarction prediction, reporting an accuracy of 82.57%. These studies highlight the utility of traditional machine learning algorithms for CAD prediction but also reveal limitations in capturing complex, non-linear interactions between features. Deep learning (DL) models, such as CNNs and recurrent neural networks (RNNs), have demonstrated significant advancements in CAD prediction tasks. Dutta et al. (2020) developed a CNN-based coronary heart disease diagnosis model, achieving near-perfect accuracy when applied to high-dimensional imaging data. Similarly, Krishnan et al. (2021)[8] utilized RNN and long short-term memory (LSTM) architectures with SMOTE to address class imbalance in the Cleveland dataset, achieving an overall accuracy of 98.5%. These approaches emphasize the superior ability of DL models to capture intricate patterns in data, particularly for time-series or imaging applications. Other studies have explored hybrid approaches, combining advanced algorithms with data augmentation techniques to improve model performance. For instance, Wang et al. (2021)[9] introduced a GAN-based method to address class imbalance, achieving an impressive 99.71% accuracy for arrhythmia detection. Similarly, Rai and Chatterjee (2021)[10] combined CNN and LSTM models with SMOTE-TomekLink to improve myocardial infarction detection. Despite these advancements, simpler models still play an important role in CAD prediction, particularly in resource-constrained settings. For example, Shah et al. (2020)[11] found that SVMs outperformed random forest and logistic regression on the Cleveland dataset, achieving a 95% accuracy rate. These findings demonstrate that while traditional algorithms can deliver strong baseline performance, they may lack the flexibility and scalability required for more complex datasets or broader clinical applications. Neural networks, particularly feedforward architectures, bridge the gap between traditional models and more computationally intensive approaches like CNNs and GANs. While they may not reach the performance levels of deep learning models in certain contexts, feedforward neural networks offer a balance of simplicity, interpretability, and predictive power. This study builds on these insights by evaluating feedforward neural networks for CAD prediction, focusing on their ability to capture non-linear relationships within the data while maintaining computational efficiency. By addressing gaps in accessibility and practicality, this research aims to demonstrate how neural networks can offer strong predictive capabilities without the extensive computational demands often associated with more complex architectures.

Dataset Description

The dataset used originates from a study that compiled 3 patient test groups[12] and is publicly available on Kaggle[13]. The dataset comprises several features relevant to CAD diagnosis including blood pressure measurements, cholesterol levels, chest pain types, resting electrocardiographic results, and the presence of exercise-induced angina. Additionally, demographic information such as age and gender are also available. The dataset contains 917 observations with 11 features and 1 label. The following is a description of each feature and label with Table 1 as an example consisting of 5 patients from the dataset.

Data Preprocessing

The dataset used in this study comprises various patient attributes, including demographic information, clinical indicators, and diagnostic test results totaling 11 features. Before constructing predictive models, several preprocessing steps are undertaken to ensure data quality and model compatibility. To handle missing values, we identify and address missing values in the dataset, particularly in variables like blood pressure and cholesterol levels, which are crucial indicators of heart health. Specifically, median imputation is used to replace missing values, as it provides a resilient approach that is less affected by outliers compared to mean imputation. Median imputation involves replacing missing values with the median value of the non-missing observations for that variable, which helps maintain the central tendency of the data without being skewed by extreme values. Categorical variables such as sex, chest pain type, resting ECG, and ST slope are encoded to facilitate model training. For binary categories like sex and exercise angina, we employ binary encoding (1 for positive, 0 for negative; 1 for M, 0 for F). Multi-class categories like chest pain type and resting ECG are transformed using one-hot encoding to represent each category as a binary feature. Continuous variables

Table 1. Five example patients in the dataset. Chest Pain Type: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]. Resting ECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria].

Feature Name & Unit	**Age** [Numeric value between 28 and 77]	**Sex** [M: Male, F: Female]	**Chest Pain Type** [TA, ATA, NAP, ASY]	**Resting BP** [mm Hg]	**Cholesterol** [mm/dl]	**Fasting BS** [1: if FastingBS > 120 mg/dl, 0: otherwise]	**Resting ECG** [Normal, ST, LVH]	**Max HR** [Numeric value between 60 and 202]	**Exercise Angina** [Y: Yes, N: No]	**Old Peak** [Numeric value measured in depression]	**ST Slope** [Up: upsloping, Flat: flat, Down: downsloping]	**CAD** [1: Has CAD, 0: Normal]
Feature description	age of the patient	sex of the patient	chest pain type	resting blood pressure	serum cholesterol	fasting blood sugar	resting electrocardiogram results	maximum heart rate achieved	Exercise induced angina	old peak = ST	the slope of the peak exercise ST segment	Tested positive or negative for CAD
Patient 1	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
Patient 2	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
Patient 3	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
Patient 4	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
Patient 5	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

are standardized using z-score normalization to ensure uniform scaling across features.

Model Architecture

With the data preprocessed and ready for modeling, we employ a feedforward neural network architecture for CAD prediction [14], comprising multiple layers of connected neurons. The choice of activation functions, layer sizes, and regularization techniques influences model performance and generalization capabilities. We experiment with several neural network architectures to develop predictive models to evaluate their effectiveness in capturing underlying patterns in the data. For all models, the input layer will be made up of 11 nodes, each representing 1 individual feature. The following figures represent the basic model architecture [15] of nodes and layers. These figures do not include detailed information such as the weights, biases, or activation functions used in the models.

Model 1, shown in Figure 1 consists of a simple architecture where the output layer is a dense layer with a single neuron, utilizing a sigmoid activation function to predict binary outcomes. The model was trained for 320 epochs. All further models also employ a single neuron with the sigmoid activation function as the output layer for the same purpose.

Model 2, shown in Figure 2, introduces more complexity with its architecture. It includes a hidden layer with 2 neurons, each using the tanh activation function to introduce non-linearity. This model was trained for 400 epochs.

Model 3, shown in Figure 3, is more advanced with two hidden layers. The first hidden layer contains 4 neurons with the rectified linear unit (ReLU) activation function, providing strong non-linear transformation capabilities. The second hidden layer has 2 neurons using the tanh activation function, which can capture more nuanced features. This model was trained for 2000 epochs. The complexity of Model 3 lies in the addition of multiple hidden layers, which significantly increases the number of parameters, making it capable of capturing more intricate relationships within the data but requires an extended training period to optimize its parameters fully.

Model 4, shown in Figure 4, builds upon the previous models with two hidden layers using the ReLU activation function. The first hidden layer has 4 neurons, and the second hidden layer comprises 2 neurons. This model was trained for 3000 epochs.

Model 5, shown in Figure 5, has a simpler architecture compared to Model 4 but still utilizes powerful components. It includes a hidden layer with 4 neurons using the ReLU activation function, which helps in learning complex patterns from the data. This model was trained for 3000 epochs.

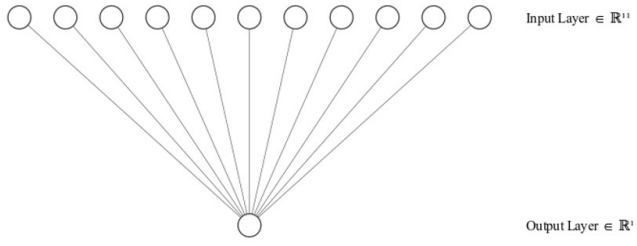

Fig. 1 Model 1 Architecture

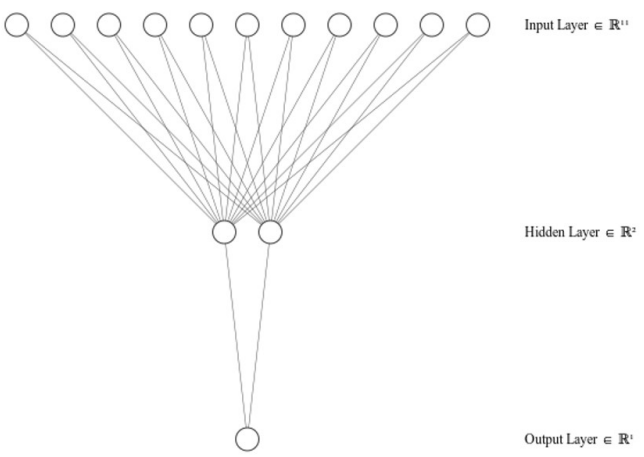

Fig. 2 Model 2 Architecture

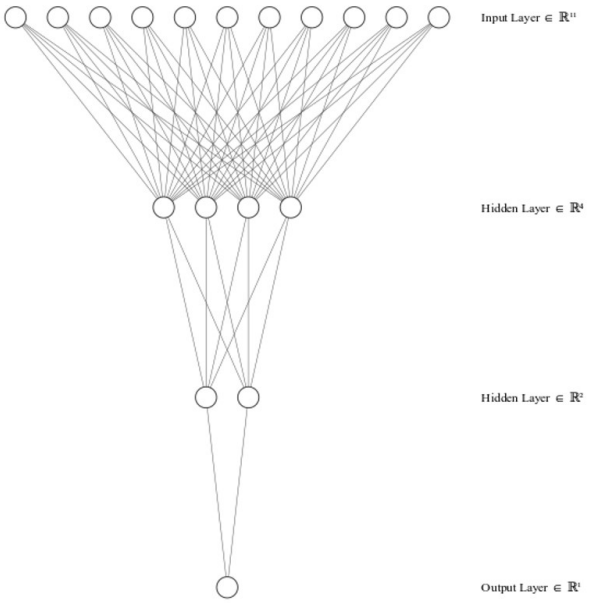

Fig. 3 Model 3 Architecture

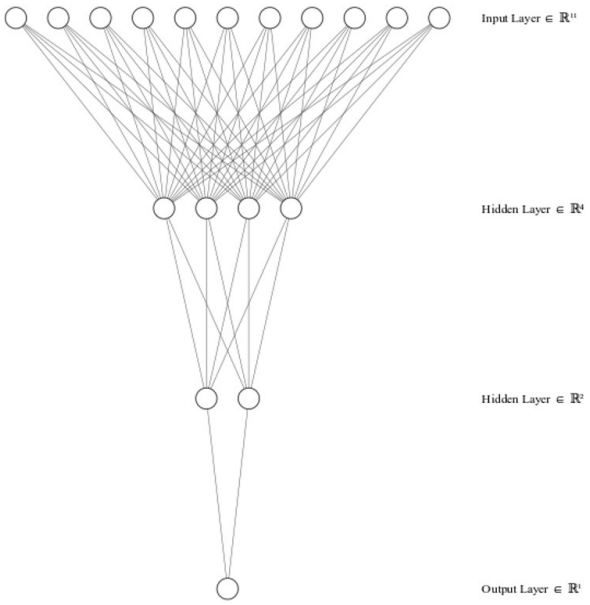

Fig. 4 Model 4 Architecture

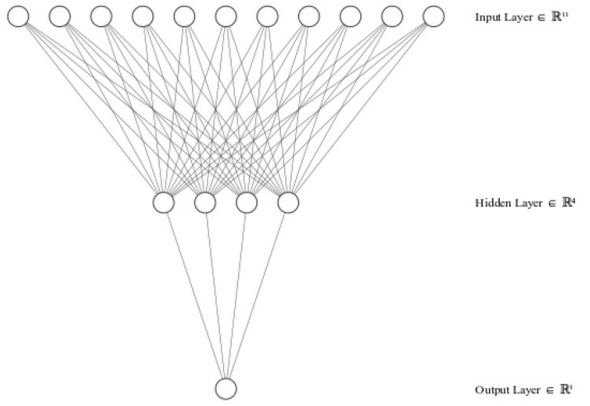

Fig. 5 Model 5 Architecture

Training and Evaluation

The dataset was divided into training and validation sets using a 75:25 split to facilitate model training and evaluation. During training, we monitor loss metrics such as binary cross-entropy and accuracy to gauge the model's effectiveness. We employ early stopping to prevent overfitting and save computational resources. Additionally, we utilize classification reports to evaluate model performance on both training and validation data, considering metrics such as precision, recall, and F1-score.

The plot for Model 1, shown in Figure 6 below, shows a steep decline in loss initially, indicating rapid learning during the early epochs. This sharp decrease reflects the model's quick grasp of the most basic patterns in the data. However, as the epochs progress, the loss curve reaches a plateau. This plateau signals that the model has exhausted its learning capacity with its simple architecture. With only one neuron and a sigmoid activation, the model can only capture fundamental patterns and is limited in its ability to model more complex relationships in the data.

In Model 2, shown in Figure 7, the loss plot shows a more gradual and sustained decline over 400 epochs. This indicates that the model, with its slightly increased complexity of two neurons and tanh activation, has a greater capacity to learn and capture nuanced patterns in the data. The slower, steady reduction in loss compared to Model 1 suggests that the model continues to learn and optimize its parameters throughout the training process. The tanh activation function allows the model to handle more complex relationships than a simple linear model,

enabling it to perform better without quickly hitting a plateau. This extended period of decreasing loss illustrates the benefits of adding a bit more complexity to the model architecture.

While the loss curves for Models 3, 4, and 5, shown in Figures 8, 9, and 10, respectively, appear to have a similar shape with a drastic initial decline, the duration over which they train (2000 and 3000 epochs) means that their learning process is far more gradual than what is observed in Models 1 and 2. However, their epochs are compressed into similarly sized graphs, making the rapid decline in their loss functions misleadingly steep when viewed on the same scale as Models 1 and 2.

The loss plot for Model 3 exhibits a prolonged and relatively gradual decrease over 2000 epochs During this period, minor fluctuations are observed in the loss values. These fluctuations are irregular rather than periodic, suggesting that they are not linked to consistent changes in the optimization process, such as learning rate oscillations. Instead, they are likely a result of the inherent randomness in the data sampling process and stochastic nature of gradient descent during training. Such fluctuations are typical for neural networks trained for extended periods, particularly when dealing with complex architectures like Model 3, which includes two hidden layers and a large number of parameters. The presence of fluctuations does not indicate convergence issues or inadequate learning rates but instead reflects the ongoing fine-tuning of parameters as the model adjusts to capture intricate patterns in the data. Furthermore, statistical analysis of these fluctuations reveals that they remain within a range of ±2% of the average validation loss, which is not significant enough to affect the model's predictive performance. Importantly, the validation accuracy remains stable despite these fluctuations, suggesting that the model maintains its generalization capability without overfitting. These observations indicate that the fluctuations are an expected part of the training process for a model of this complexity. However, the validation accuracy plateaus after approximately 1500 epochs, suggesting that the model begins to overfit the training data. This behavior indicates that the model is focusing on learning intricate patterns specific to the training data rather than generalizable features applicable to unseen data. The overfitting is further evidenced by the widening gap between the training and validation loss curves during the latter epochs.

For Model 4, the loss plot shows a rapid decline over an extended training period of 3000 epochs. The validation accuracy and training loss do not improve significantly beyond 1000 epochs. This extended training period, combined with minimal gains in validation performance, suggests that the model's higher complexity (two hidden layers with ReLU activation) may be introducing unnecessary parameters that capture noise rather than meaningful patterns.

The loss plot for Model 5 reveals an interesting pattern. The training loss decreases steadily during the initial stages of training but plateaus at approximately 1250 epochs. From this point onward, the training loss remains consistent, showing no further decrease until the end of the 3000 training epochs. This plateau suggests that the model has reached its optimal learning capacity and is no longer improving its fit to the training data. The early plateauing of the training loss indicates that Model 5's simpler architecture, with a single hidden layer of four neurons, prevents the model from overfitting by limiting its ability to capture overly complex patterns or noise within the training data. This behavior contrasts with the more complex Models 3 and 4, where the training loss declines while validation performance stagnates, signaling overfitting. The stability of the training and validation losses after 1250 epochs highlights Model 5's ability to generalize effectively. The lack of further improvement in training loss after this point suggests that the model has successfully identified the core patterns in the data without over-parameterization.

Regularization was applied in the form of early stopping to prevent excessive overfitting. For example, training was halted once validation error ceased to improve significantly, as observed in Model 5 after 1250 epochs. Additional techniques, such as dropout and weight decay, could be explored in future work to enhance the models' robustness further. These approaches might help narrow the training and validation performance gap, particularly for complex models like Models 3 and 4.

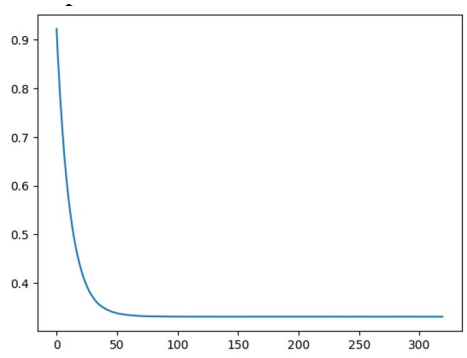

Fig. 6 Model 1 Training Loss

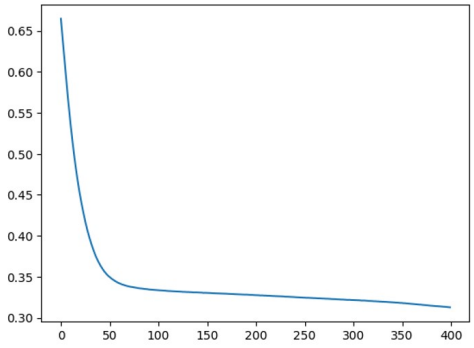

Fig. 7 Model 2 Training Loss

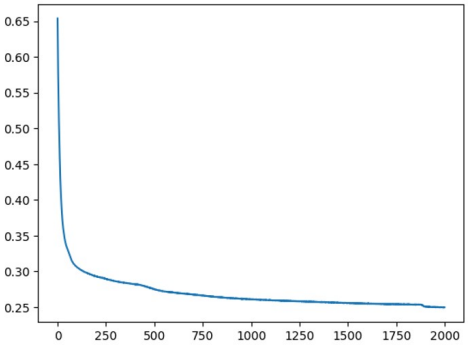

Fig. 8 Model 3 Training Loss

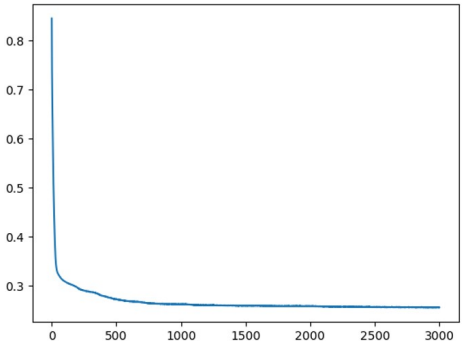

Fig. 9 Model 4 Training Loss

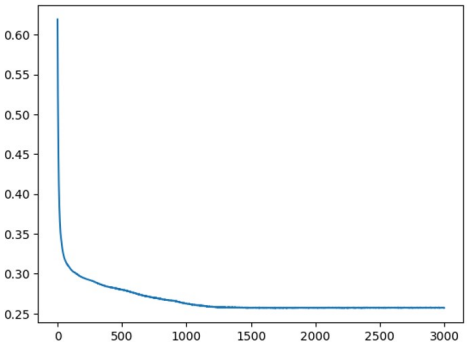

Fig. 10 Model 5 Training Loss

Results and Discussion

In our studies, We observe variations in model performance based on architecture complexity, activation functions, and preprocessing techniques. While simpler models tend to generalize better, more complex architectures may risk overfitting, necessitating careful regularization. Our experiments reveal that neural network models trained on the dataset achieve promising results in predicting CAD, as shown in Table 2 and Figure 11.

Table 2 presents validation error, training error, and the average of both errors for each model. Validation error indicates the model's ability to generalize to unseen data, while training error reflects the model's fit to the training dataset. Lower validation errors suggest better generalization, which is critical for reliable predictive performance in real-world applications. In contrast, lower training errors indicate strong learning from the training data. The average error serves as a summary metric to assess overall model performance, balancing the training and validation errors. This metric is particularly useful in comparing models by providing a single value that reflects both their fit and generalization capabilities.

As shown in Table 2, validation errors are consistently higher than training errors for all models, indicating potential overfitting. Model 3, despite achieving the lowest training error of 0.242, exhibits the highest validation error of 0.445, demonstrating its tendency to overfit. This disparity arises from the model's complexity, which includes two hidden layers and numerous parameters, enabling it to memorize training data but struggle with generalization. Regularization techniques such as dropout or weight decay could have mitigated this issue by discouraging the model from relying too heavily on specific patterns in the training data.

The average errors across models, as shown in Table 2, provide an overall performance comparison. Model 5 achieves the lowest average error of 0.308, reflecting its superior balance between training and validation performance. The differences between average errors for other models, such as Model 3 (0.344) and Model 2 (0.343), indicate that these models are less effective at generalizing, likely due to their higher and lower complexity, respectively. While statistical significance was not formally assessed, the trends in average errors suggest that a moderately complex architecture like Model 5 offers a more reliable solution for CAD prediction.

Model 1 demonstrates relatively balanced and consistent performance across both the training and validation datasets. It achieves a validation error of 0.342 and a training error of 0.330, suggesting a good fit to the data with minimal signs of overfitting. The model's precision, recall, and F1-scores for both classes hover around 0.87, which contributes to an overall accuracy of 87%. This uniformity in performance metrics indicates that Model 1 is well-calibrated and effectively generalizes to unseen data, a crucial trait for predictive modeling. The precision for Model 1 is consistent between the training and validation sets, showing that the model performs similarly well across different datasets.

Model 2 shows a slight improvement in training error, achieving 0.312, but maintains a comparable, albeit slightly worse, validation error of 0.373. The model's precision, recall, and F1-scores are slightly higher, particularly for the positive class (CAD output '1'), leading to an overall accuracy of 88%. This indicates that Model 2 generalizes well and avoids overfitting, given the close alignment of its training and validation errors. Despite the slight increase in validation error compared to Model 1, the improved precision and recall for the positive class enhance the model's reliability in predicting true positives of CAD.

Table 2. Validation and Training Error of each Model

Error Type	Model 1	Model 2	Model 3	Model 4	Model 5
Validation Error (BCE)	0.342	0.373	0.445	0.434	0.359
Training Error (BCE)	0.330	0.312	0.242	0.254	0.256
Average of Both Errors (BCE)	0.336	0.343	0.344	0.344	0.308

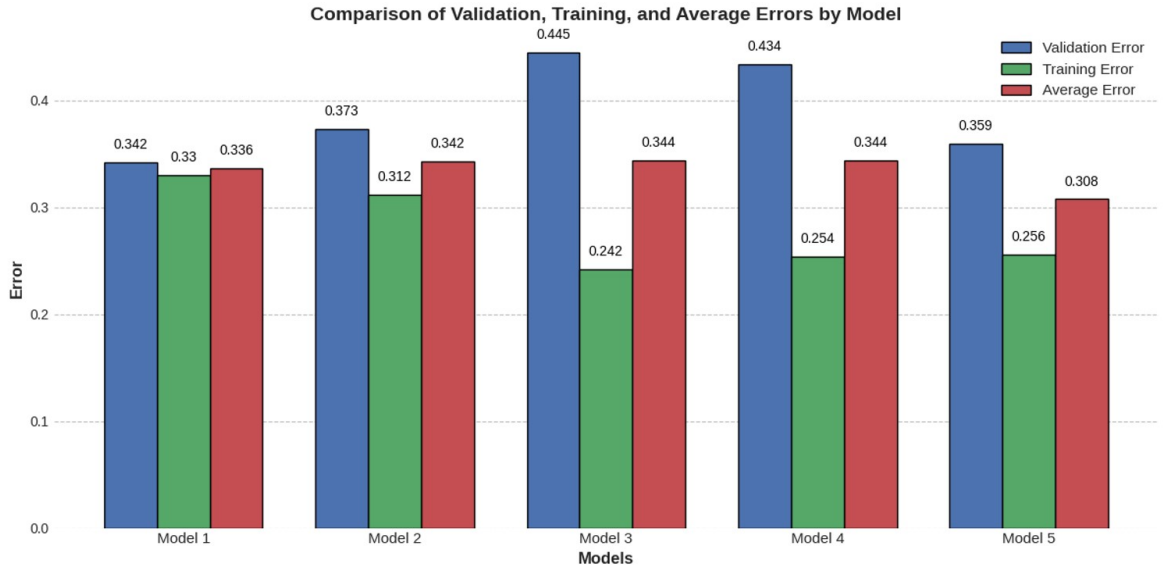

Fig. 11 Graph representing Validation, Training, and Average Error in Table 2

Table 3. Training and Validation Set Classification Report of Model 1

Classification Set	Precision		Recall		f1-score	
	Training	Validation	Training	Validation	Training	Validation
0	0.87	0.85	0.85	0.84	0.86	0.84
1	0.87	0.88	0.89	0.89	0.88	0.88
Macro avg	0.87	0.86	0.87	0.86	0.87	0.86
Weighted avg	0.87	0.87	0.87	0.87	0.87	0.87

This performance suggests that Model 2 may be better suited for applications where correctly identifying positive instances is critical.

Model 3 highlights significant overfitting issues, achieving a notably low training error of 0.242 but a significantly higher validation error of 0.445. While the model's training performance is exceptionally high, with an accuracy of 94%, this drops sharply to 84% on the validation set. The precision, recall, and F1-scores are very high for the training set but noticeably lower for the validation set, indicating that the model may be overly complex and not generalize well to unseen data. This disparity between training and validation performance emphasizes the model's tendency to overfit, meaning it has learned the training data too well, including noise and minor fluctuations that do not generalize to new data. This significant overfitting makes Model 3 less suitable for practical applications despite its high training performance, as its predictions on new data are less reliable.

Noise in the dataset likely stems from patient variability, measurement inaccuracies, and minor outliers. For example, variations in blood pressure readings may result from temporary stress or physiological factors unrelated to CAD, while measurement errors could arise from inconsistent testing equipment or methods. Such noise can obscure true patterns in the data, challenging the model's ability to differentiate between relevant and irrelevant features. Addressing noise through techniques such as robust preprocessing and regularization helps improve model accuracy and generalization.

Model 4 also presents signs of overfitting, as evidenced by its low training error of 0.254 but a higher validation error of 0.434. The model achieves an accuracy of 90% on the training set, which drops to 86% on the validation set. Although the training set shows high precision, recall, and F1-scores, these metrics are lower on the validation set, particularly in terms of precision. This suggests that the model fits the training data

Table 4. Training and Validation Set Classification Report of Model 2

Classification Set	Precision		Recall		f1-score	
	Training	Validation	Training	Validation	Training	Validation
0	0.89	0.83	0.84	0.84	0.86	0.83
1	0.87	0.88	0.91	0.87	0.89	0.87
Macro avg	0.88	0.85	0.88	0.85	0.88	0.85
Weighted avg	0.88	0.86	0.88	0.86	0.88	0.86

Table 5. Training and Validation Set Classification Report of Model 3

Classification Set	Precision		Recall		f1-score	
	Training	Validation	Training	Validation	Training	Validation
0	0.94	0.81	0.92	0.81	0.93	0.81
1	0.94	0.86	0.95	0.86	0.94	0.86
Macro avg	0.94	0.84	0.94	0.83	0.94	0.84
Weighted avg	0.94	0.84	0.94	0.84	0.94	0.84

too closely, capturing specific patterns and noise that do not generalize to new data effectively. The decline in validation performance highlights that the model's complexity might be too high for the dataset, leading to overfitting. This performance indicates that Model 4 may not be the best choice for tasks requiring robust generalization to new data, despite its strong training performance.

Model 5 manages to reduce overfitting compared to Models 3 and 4, likely due to its less complex architecture, achieving a training error of 0.256 and a validation error of 0.359. This smaller parameter count limits the model's capacity to overfit to noise or irrelevant patterns in the training data. By focusing on core data patterns, Model 5 strikes a balance between learning and generalization, making it more effective for this dataset. The model maintains an accuracy of 90% for the training set and 84% for the validation set. While it exhibits high precision and recall scores on the training data, these metrics are slightly lower on the validation data. This suggests that Model 5 strikes a better balance between fitting the training data and generalizing to validation data compared to the more overfit Models 3 and 4. The reduced gap between training and validation errors indicates improved generalization and robustness, making Model 5 a more reliable option for practical applications. However, the recall drops slightly to 0.81, while precision remains relatively stable at 0.86. This suggests that Model 5 is better at predicting true negatives (correctly identifying non-CAD cases) than true positives (identifying CAD cases). This is significant for CAD prediction models, as higher precision reduces false alarms, which is critical in clinical settings to avoid unnecessary interventions. However, the slight decrease in recall highlights a potential area for improvement, as failing to identify CAD cases could lead to missed diagnoses. Adjusting the decision threshold or incorporating additional features might help mitigate this imbalance.

The results highlight that while complex architectures like Models 3 and 4 can capture intricate patterns in the data, their increased depth and parameter count make them prone to overfitting, reducing their ability to generalize effectively. In contrast, Model 5 demonstrates that simpler architectures can better balance learning and generalization. This reveals the importance of selecting appropriately scaled models for datasets of this size and complexity or incorporating regularization techniques to mitigate overfitting.

Practical Implications

Deploying CAD prediction models in clinical environments carries significant potential and critical challenges. Models that perform well in controlled datasets must be rigorously evaluated to ensure they meet the practical requirements of clinical settings. This includes reliability, interpretability, and integration into existing healthcare workflows. Clinicians must understand the rationale behind a model's predictions. Tools such as feature importance analysis or explainable AI methods could help bridge the gap between the technical complexity of neural networks and clinical decision-making. Techniques like SHAP (Shapley Additive Explanations) or LIME (Local Interpretable Model-agnostic Explanations) could make the model's predictions more transparent [16], creating a more trustworthy diagnosis among clinicians.

Generalization is critical to CAD prediction models, particularly in clinical settings where consistent and reliable performance is essential for patient care. Models that fail to generalize effectively are prone to overfitting, learning patterns specific to the training data, including noise, rather than capturing underlying trends that apply to unseen data. Overfitting can lead to two significant risks in practical applications:

1. False Positives (Incorrectly Identifying Healthy Patients as At-Risk): An overfitted model may flag healthy individuals

Table 6. Training and Validation Set Classification Report of Model 4

Classification Set	Precision		Recall		f1-score	
	Training	Validation	Training	Validation	Training	Validation
0	0.90	0.84	0.88	0.84	0.89	0.84
1	0.90	0.88	0.92	0.88	0.91	0.88
Macro avg	0.90	0.86	0.90	0.86	0.90	0.86
Weighted avg	0.90	0.86	0.90	0.86	0.90	0.86

Table 7. Training and Validation Set Classification Report of Model 5

Classification Set	Precision		Recall		f1-score	
	Training	Validation	Training	Validation	Training	Validation
0	0.91	0.82	0.88	0.81	0.89	0.81
1	0.90	0.86	0.93	0.87	0.91	0.86
Macro avg	0.90	0.84	0.90	0.84	0.90	0.84
Weighted avg	0.90	0.84	0.90	0.84	0.90	0.84

as being at risk for CAD, resulting in unnecessary diagnostic tests, increased healthcare costs, and patient anxiety. This strains healthcare resources and undermines trust in the predictive model.

2. False Negatives (Failing to Identify At-Risk Patients): More concerning, an overfitted model may fail to identify patients genuinely at risk for CAD, leading to missed opportunities for early intervention and potentially severe health consequences. In clinical practice, this limitation could compromise patient safety and outcomes.

Furthermore, overfitted models often exhibit variability in their predictions when applied to slightly different datasets or populations. For instance, a model trained on data predominantly from one demographic group may perform poorly when deployed in a different demographic context. Such variability reduces the reliability of the model and its applicability to diverse patient populations, making it unsuitable for widespread clinical use.

To mitigate these risks, designing models with generalization capabilities is imperative. This includes employing regularization techniques, optimizing hyperparameters, and using diverse, representative datasets during training. External validation on independent datasets can provide a clearer picture of the model's ability to generalize across populations. By prioritizing generalization, CAD prediction models can deliver consistent, reliable, and clinically actionable results, ultimately improving patient care and healthcare efficiency.

Study Limitations

This study has several limitations that should be addressed to improve the robustness and generalizability of the models. First, the dataset used in this study was relatively small, comprising only 918 patient data rows. While this provided valuable insights into the model's performance, the limited sample size may reduce the generalizability of the findings, mainly when applied to more extensive and more diverse populations. Smaller datasets can also increase the risk of overfitting, as models might capture patterns specific to the sample rather than generalizable trends.

Second, the study employed only one regularization technique, early stopping, to mitigate overfitting. While this approach helped prevent excessive training beyond the point of diminishing returns, other regularization methods, such as L1 or L2 regularization, dropout, or weight decay, could have been explored. These techniques can further discourage models from relying too heavily on specific features and enhance their ability to generalize new data.

Third, preprocessing techniques were limited to replacing missing values with median imputation. Although this is a simple and effective method, more advanced techniques like K-nearest neighbors (KNN) imputation could have been employed to capture relationships between features better and reduce potential biases introduced by imputation. Additionally, SMOTE (Synthetic Minority Oversampling Technique) could have been applied to address class imbalance in the dataset, mainly if the number of CAD-positive cases was significantly smaller than that of negative cases. This would have improved the model's ability to predict true positives and reduced bias toward the majority class[5].

Finally, biases in the dataset present a notable limitation. The dataset may be overrepresented by specific patient demographics (e.g., age groups, genders, or socioeconomic backgrounds) or clinical conditions, leading to skewed predictions that are not representative of broader populations. For instance, if the dataset included more cases from a single demographic group, the model might perform better for that group while underperforming for others. Additionally, bias can stem from measure-

ment inaccuracies or the inclusion of features that are proxies for demographic variables, unintentionally embedding disparities into the model. These biases affect model performance and raise ethical concerns when deploying predictive models in clinical settings.

Future work should address these limitations using larger, more diverse datasets and implementing a broader range of regularization and preprocessing techniques. Exploring methods to identify and mitigate dataset biases, such as fairness-aware machine learning, and conducting external validation on independent datasets would enhance the reliability and applicability of the models.

Conclusion

In this paper, we demonstrate the efficacy of neural networks in predicting CAD using patient data. Our analysis highlights the importance of data preprocessing, model architecture selection, and evaluation metrics in developing accurate and generalizable predictive models. We observe variations in model performance based on architecture complexity, activation functions, and preprocessing techniques. While simpler architectures achieve competitive performance, deeper models require careful regularization to prevent overfitting [3,4,17]. Several strategies can be implemented to enhance our model development and address the issues encountered in our results. To combat overfitting, we can introduce regularization methods such as L1 or L2 regularization [2–4] to discourage the model from learning overly complex patterns. Another effective approach is applying dropout during training [2], which forces the network to learn more redundant representations by randomly dropping a fraction of neurons. Additionally, implementing early stopping can prevent the model from overfitting [18] by halting training once the validation performance ceases to improve. Employing cross-validation [2–4] instead of a single train-validation split would provide a more reliable estimate of model performance, ensuring that the model's generalization capabilities are evaluated across multiple data subsets. Simplifying the model by pruning unnecessary neurons and layers can also help reduce overfitting by retaining only the most significant connections [4]. It is important to acknowledge the substantial role of data quantity in achieving optimal model performance [3,4,17]. To further enhance CAD prediction, future work can incorporate additional data sources, such as genetic information or imaging data from a Coronary angiography, to improve model performance potentially with the cost of accessibility. Additionally, investigating alternative neural network architectures or exploring different learning algorithms could be valuable avenues for further research.

Acknowledgments

I would like to express my sincere gratitude to Joanna Gilberti and Professor Guillermo Goldzstein for their invaluable support and guidance throughout this research. Their insights and teaching were instrumental in shaping this study.

References

1 J. Brown, T. Gerhardt and E. Kwon, *Risk factors for coronary artery disease*, https://www.ncbi.nlm.nih.gov/books/NBK554410/, Retrieved June 19, 2024, from.

2 T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*.

3 F. Azuaje, I. Witten and E. Frank, *Data mining: Practical machine learning tools and techniques*.

4 I. Witten and E. Frank, *Data mining: Practical machine learning tools and techniques*.

5 M. Ahsan and Z. Siddique, *Machine learning-based heart disease diagnosis: A systematic literature review*, https://api.semanticscholar.org/CorpusID:245124466., Retrieved June 19, 2024, from.

6 S. Bashir, U. Qamar and F. Khan, *A multicriteria weighted vote-based classifier ensemble for heart disease prediction*.

7 A. Daraei and H. Hamidi, *An efficient predictive model for myocardial infarction using cost-sensitive J48 model*.

8 S. Krishnan, P. Magalingam and R. Ibrahim, *Hybrid deep learning model using recurrent neural network and gated recurrent unit for heart disease prediction*.

9 Y. Wang, L. Sun and S. Subramani, *CAB: Classifying arrhythmias based on imbalanced sensor data*.

10 H. Rai and K. Chatterjee, *Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data*.

11 D. Shah, S. Patel and S. Bharti, *Heart disease prediction using machine learning techniques*.

12 R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee and V. Froelicher, *International application of a new probability algorithm for the diagnosis of coronary artery disease*.

13 fedesoriano, *Heart failure prediction dataset*, https://www.kaggle.com/fedesoriano/heart-failure-prediction, Retrieved June 16, 2024, from.

14 J. Schmidhuber, *Deep learning in neural networks: An overview*.

15 A. Lenail, *Neural network SVG: Interactive visualization tool for neural networks*, https://alexlenail.me/NN-SVG/index.html, Retrieved June 19, 2024, from.

16 K. E. AI, I. LIME and S. Kaggle, https://www.kaggle.com/code/khusheekapoor/explainable-ai-intro-to-lime-shap., Retrieved June 19, 2024, from.

17 P. Domingos, *A few useful things to know about machine learning*.

18 T. Zhang and B. Yu, *Boosting with early stopping: Convergence and consistency*.

