A Supervised Deep Learning Model for the Detection of Cardiovascular Disease

Ananya Saridena, Abhaya Saridena, and Jothsna Kethar

ABSTRACT

In our world today, cardiovascular disease (CVD) stands as the foremost cause of death worldwide, claiming the lives of nearly 20 million individuals annually. As CVD continues to burden the healthcare industry, there is a critical need for early detection and prevention. The rise of Artificial Intelligence in the medical field offers a range of capable solutions. In order to address the problem, this paper presents the development of a simple, supervised deep learning model for detecting cardiovascular disease in patients. The research focused on creating a model enriched with multiple layers, activation functions, optimizers, and loss functions. The chosen approach leveraged the power of AI to analyze labeled patient data and map input features to corresponding class labels, enabling accurate detection of CVD. The dataset used contained 70,000 patient records with 12 different clinical attributes. In addition, it provides an overview of the most common types of cardiovascular disease, such as coronary artery disease, aortic valve disease, stroke and peripheral artery disease. The accuracy of the obtained results from the deep learning model was up to 73%. The utilization of AI systems can present a novel approach to addressing daily challenges within the rapidly-evolving world of medicine. Health personnel can take advantage of rapidly changing artificial intelligence and user friendly deep learning models to detect similar future medical concerns.

Introduction

Cardiovascular disease (CVD) encompasses a large range of heart and blood vessel disorders and remains the leading cause of death worldwide. Despite advancements in treatment, urgency necessitates more advanced diagnostic approaches. Supervised machine learning techniques have emerged as invaluable tools for interpreting and predicting heart disease onset by analyzing labeled patient data. The algorithms used in these machine learning models can map input features to corresponding class labels, enabling accurate predictions for new data instances. Early diagnosis of CVD allows for timely interventions and lifestyle modifications to mitigate the risk of life-threatening events. This study aims to develop a predictive system for cardiovascular disease using historical patient data, ensuring high accuracy and compatibility with connected healthcare systems. Through the use of deep learning neural networks, the efficacy of machine learning models will be systematically assessed. The subsequent sections provide an explanation of key information regarding cardiovascular disease including risk factors, specific types of CVD, and current diagnostic techniques before explaining detailed accounts of the methods used to create the model and discussing produced results.

Cardiovascular Disease

Cardiovascular Disease encompasses various conditions that affect the body's cardiovascular system, responsible for heart and blood vessel functions. This essential system ensures nutrient and oxygen transport to all organs in the human body. As such, maintaining its health is crucial for overall well-being. Early detection and timely intervention of CVD can help prevent complications or long-term damage.
Modifiable Risk Factors

Cardiovascular disease is a significant global health challenge. Fortunately, CVD is closely related to “behavioral risk factors.” This means that CVD is primarily related to a person’s changeable lifestyle choices with the major risk factors including poor diet/nutrition, physical inactivity, alcohol consumption, tobacco use, obesity, and hypertension, also known as high blood pressure. As such, making particular lifestyle changes can inevitably reduce a patient’s risk of contracting CVD. Avoiding alcohol and tobacco usage, adopting a low sodium, balanced and nutritious diet, and ensuring regular physical activity are proven to reduce the risk of cardiovascular disease. In addition, drug treatment of diabetes, hypertension, and high blood lipids can act to prevent heart attacks and strokes, as administered on the diagnosis of a licensed medical professional.

Non-Modifiable Risk Factors

Although a majority of cardiovascular risk factors can be categorized as changeable lifestyle or behavioral factors, there are also several factors increasing a patient’s risk of contracting cardiovascular disease that are not removable. The three main non-modifiable contributing factors of CVD are family history, ethnicity, and age.

Certain genetic tendencies have been found to raise the risk of getting cardiovascular disease. For instance, blood clotting tendencies have genetic traits that can contribute to atherosclerosis, increased blood cholesterol levels, inflammation, etc. may be among these hereditary predispositions. In addition, hereditary disorders may also be directly linked to a higher chance of developing CVD. Hypercholesterolemia is one example of this. Unhealthy cholesterol levels, which are typically a controllable risk factor, cannot be eliminated or reduced due to this genetic abnormality. Healthcare providers must also consider family ancestry and ethnicity. Patients descending from particular regions including parts of Africa, the Caribbean, and South Asia have been found to carry a higher inclination towards. However, the exact reason for this increase in risk exists has not been confirmed scientifically. Age is the final non-modifiable risk factor. The physiology of the body inevitably changes as we age while particular changes in the heart and blood arteries can directly increase the risk of CVD. The increasing stiffening of the myocardium and flexibility loss in blood vessels are a couple examples of physiological changes that occur in the body over time. These changes lead to less efficient blood pumping and oxygen delivery to the body’s organs and tissues. These unchangeable characteristics invariably point to a patient having a higher chance of developing CVD.

Common Types of Cardiovascular Disease

There are several types of cardiovascular disease with vastly differing implications and attributes. The subsequent sections outline the main forms of CVD:

*Coronary Artery Disease*

Coronary Artery Disease (CAD) is the singular cardiovascular disease causing the highest death toll. It stands to be one of the most prevalent forms of CVD, globally. It results from atherosclerosis in the coronary arteries, which supply blood to the heart. Atherosclerosis refers to a condition where a substance known as plaque builds in artery walls. This leads to thicker and harder arterial linings that make the passage of blood more difficult. This also leads to a high potential for eventual formation of blood clots contributing significantly towards heart attack or stroke incidents, if left untreated. Hence taking measures towards prevention while continuously monitoring any symptoms related to CAD with healthcare professionals would prove crucial.
Aortic Valve Disease

This condition affects the aortic valve in the heart. The aortic valve is not able to close properly, which results in blood leaking backwards into the aorta. The disease brings about either of two results - aortic stenosis, where the valve narrows, or aortic regurgitation when it leaks. Patients with this disease suffer symptoms including chest pain, shortness of breath, and fatigue. If left unchecked, it can lead to severe conditions such as heart failure or sudden cardiac death. Treatment options could range from medication to invasive surgeries aimed at correcting damaged valves.

Figure 1. A visual comparison of arterial conditions during atherosclerosis to normal conditions. Created and copyrighted by Ananya and Abhaya Saridena.

Figure 2. A depiction of the aortic valve during aortic regurgitation. Created and copyrighted by Ananya and Abhaya Saridena.
**Stroke**

Hemorrhagic strokes and ischemic strokes are the two primary forms of strokes. Ischemic strokes occur more frequently, resulting from a blood vessel supplying the brain being obstructed due to the development of a blood clot. This blockage causes rapid death of brain cells that oversee essential activities like walking or speaking, leading to temporary or permanent damage.

Hemorrhagic stroke, unlike ischemic stroke where a blood clot blocks the brain’s blood vessels, happens when bleeding occurs in the brain due to a ruptured blood vessel. This condition can result from uncontrolled hypertension or high blood pressure. Every moment counts during an ongoing stroke as millions of precious brain cells perish irreversibly. The key is early detection and timely treatment for better chances of recovery since cell regeneration after a stroke is impossible. But prevention is always better than cure; hence, we must prioritize avoiding risk factors associated with strokes that can cause permanent damage.

![Types of Stroke](image)

**Figure 3.** This figure depicts an overview of the two different types of strokes. Created and copyrighted by Ananya and Abhaya Saridena.

**Peripheral Artery Disease**

Peripheral arterial disease (PAD), is a common condition affecting blood circulation. It causes narrowing or blockages in the blood vessels outside of the heart and brain, mainly those that supply blood to the legs and feet. This disorder also results from atherosclerosis. Over time, these deposits can narrow arteries and restrict normal blood flow, leading to further problems such as stroke or heart failure.

**Current Diagnostic Techniques**

Currently, cardiovascular disease (CVD) can be identified via electrocardiograms (ECGs), echocardiography, and coronary angiography. These diagnostic techniques require specialized tools and experienced personnel and take a considerable amount of time to produce results. Moreover, these methods might not catch CVD’s early warning signals, delaying therapy and producing less favorable outcomes.

Through the development of quicker, more precise, and more accessible techniques, artificial intelligence (AI) has the potential to completely change the way CVD is diagnosed. In order to find trends and estimate the risk of CVD, AI algorithms may scan vast volumes of data from numerous sources, such as patient records, genetic data, and...
medical pictures. This can assist medical professionals in making an earlier diagnosis of CVD and creating individualized treatment programs for their patients. Additionally, AI can assist in overcoming some of the drawbacks of existing diagnostics.

Utilizing Deep Learning for Cardiovascular Disease Detection

The project aims to develop an accessible- and low-cost solution using advanced AI techniques for detecting cardiovascular disease at an early stage. The materials employed are readily available- and inexpensive, even in the developed world. With just a computer, access to the Internet, and a WiFi connection, an AI model is created to predict cardiovascular diseases efficiently. The advantage of using AI is its vast potential and ease of implementation capabilities that enables numerous possibilities with harnessing artificial intelligence in solving health-related issues like human heart illnesses.

At its core, artificial intelligence is a simple concept. AI models utilize patterns from algorithms or datasets to make predictions. In the context of cardiovascular disease, relevant data like patient profiles and medical records help these models understand the patterns and predict the likelihood of developing such ailments.

Choosing between machine learning (ML) and deep learning (DL) for the model was a significant decision to make. Machine learning requires explicit feature engineering, where the programmer defines the relationships between different features, unlike in deep learning, in which this process is automated. Machine learning models tend to be challenging to design but simpler to comprehend and manage, due to feature engineering. On the contrary, deep learning offers a more streamlined approach for creating models, although it can be more challenging to interpret the internal workings of the model, often referred to as the “black box” phenomenon. This means that it is more difficult to explain how a deep learning model is able to come to its conclusions. The programming language

Given the intended objectives of this investigation, the deep learning approach was chosen due to its straightforwardness. This particular method is known for its efficiency in handling intricate patterns, which is a perfect match for the CVD detection model that aims to be both quick and precise. For the purposes of this particular research the mathematical intricacies of an artificial intelligence model are not explored in depth. The purpose of this paper is to introduce a guide to producing simple, quick, and accessible models. Additionally, research also shows that deep learning neural networks, like their precursors, support vector machines, are aided by the same types of engineered features. While the basics of artificial intelligence rely on advanced mathematical knowledge, it is not always needed for developing simple models.

Overfitting and Underfitting

Overfitting and underfitting pose challenges that can impact the accuracy of artificial intelligence models. Overfitting occurs when models become overly tailored to the training set, capturing noise and inconsistencies instead of the underlying trend. While excelling on the training data, overfitted models struggle to generalize to new data, resulting in poor performance during validation. The ultimate objective is to minimize validation loss, but overfitting undermines this goal. Conversely, underfitting involves models that are too simplistic, failing to capture relevant patterns and yielding suboptimal performance in both training and validation. The key lies in striking a balance, where the model is neither overly specific nor too general to the training set, thus achieving minimized validation loss.
**Figure 5.** A graphical representation of underfitting and overfitting in machine learning models. Created and copyrighted by Ananya and Abhaya Saridena.

Each dot represents a data point used for training the model. Underfitted models produce inaccurate results that deviate from the training data, while overfitted models tightly fit the training data, including noise and outliers, but struggle with new data. An ideal model finds the middle ground, accurately capturing the underlying trends without being swayed by noise or oversimplifying the patterns.

**Methods**

**Dataset**

This study utilized the Cardiovascular Disease dataset by Svetlana Ulianova. The dataset was retrieved from kaggle and uploaded to a github repository. The dataset includes 70,000 patient records with 12 main attributes, as depicted in Table 1.
Table 1. In this table, the various attributes of data as well as classification information is provided.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute Type</th>
<th>Variable Name</th>
<th>Value Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Objective</td>
<td>age</td>
<td>int (days)</td>
</tr>
<tr>
<td>Height</td>
<td>Objective</td>
<td>height</td>
<td>int (cm)</td>
</tr>
<tr>
<td>Weight</td>
<td>Objective</td>
<td>weight</td>
<td>float (kg)</td>
</tr>
<tr>
<td>Gender</td>
<td>Objective</td>
<td>gender</td>
<td>categorical code: 1 - women, 2 - men</td>
</tr>
<tr>
<td>Systolic blood pressure</td>
<td>Examination</td>
<td>ap_hi</td>
<td>int</td>
</tr>
<tr>
<td>Diastolic blood pressure</td>
<td>Examination</td>
<td>ap_lo</td>
<td>int</td>
</tr>
<tr>
<td>Cholesterol</td>
<td>Examination</td>
<td>cholesterol</td>
<td>1: normal, 2: above normal, 3: well above normal</td>
</tr>
<tr>
<td>Glucose</td>
<td>Examination</td>
<td>gluc</td>
<td>1: normal, 2: above normal, 3: well above normal</td>
</tr>
<tr>
<td>Smoking</td>
<td>Subjective</td>
<td>smoke</td>
<td>binary</td>
</tr>
<tr>
<td>Alcohol intake</td>
<td>Subjective</td>
<td>alco</td>
<td>binary</td>
</tr>
<tr>
<td>Physical activity</td>
<td>Subjective</td>
<td>active</td>
<td>binary</td>
</tr>
<tr>
<td>Presence or absence of cardiovascular disease</td>
<td>Target Variable</td>
<td>cardio</td>
<td>binary</td>
</tr>
</tbody>
</table>

These attributes are then processed, and patterns or features are identified by the model. The correlations between features and patterns contributing to the “learned” knowledge of the model and diagnoses/decisions cannot be accessed. However, data processing and analytics can help us understand these correlations.

Developing the Dataset

The first step in any deep learning endeavor is to obtain data. The data used in the project was transformed into a CSV file and uploaded to a Github repository. Typically, one saves the converted data on their personal computer and provides its directory path to the program. But, if either lack of space or collaborative sharing online prevent the storage of said file locally, the program would be futile. Instead, the program has code that takes a raw Github link and reads it directly. The program will not work if the URL of the repository is not pasted. Though this may seem more complicated than the file path method, the advantage of using a raw Github link is that anyone with access to the Github repository can run the model regardless of computer memory limitations.

The next step is to preprocess the data. First, the values of the age column were rounded from days to years. Then, to make processing simpler, gender was split into two different columns: female and male. Both columns used binary values to indicate gender for each patient (1 == true, 0 == false). Finally, Body Mass Index (BMI) is calculated from patient weight and height values using the following line of code (the formula):
df.insert(5, 'bmi', round((df['weight'] / (df['height'] / 100) ** 2), 2))

These values are then assigned to a new column named BMI.

<table>
<thead>
<tr>
<th></th>
<th>age</th>
<th>female</th>
<th>male</th>
<th>height</th>
<th>weight</th>
<th>bmi</th>
<th>ap_hi</th>
<th>ap_lo</th>
<th>cholesterol</th>
<th>gluc</th>
<th>smoke</th>
<th>alco</th>
<th>active</th>
<th>cardio</th>
<th>years</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50.36</td>
<td>0</td>
<td>1</td>
<td>168</td>
<td>62.00</td>
<td>21.97</td>
<td>110</td>
<td>80</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>55.38</td>
<td>1</td>
<td>0</td>
<td>156</td>
<td>85.00</td>
<td>34.93</td>
<td>140</td>
<td>90</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>51.83</td>
<td>1</td>
<td>0</td>
<td>165</td>
<td>64.00</td>
<td>23.51</td>
<td>130</td>
<td>70</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>51</td>
</tr>
<tr>
<td>3</td>
<td>48.25</td>
<td>0</td>
<td>1</td>
<td>169</td>
<td>82.00</td>
<td>28.71</td>
<td>150</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>47.84</td>
<td>1</td>
<td>0</td>
<td>156</td>
<td>56.00</td>
<td>23.01</td>
<td>100</td>
<td>60</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47</td>
</tr>
</tbody>
</table>

Figure 6. This image is an excerpt of the dataset following preprocessing. Created and copyrighted by Ananya and Abhaya Saridena.

After preprocessing, the following points were identified regarding the dataset:
1. The mean age for patients is 53
2. The percentage of males is 35
3. The percentage of females is 65
4. The percentage of smokers is 8
5. The percentage of alcoholics is 5
6. The percentage of patients who are active is 80
Figure 7. This figure demonstrates the relationship between age and the presence of cardiovascular disease within the dataset. The dataset reveals that patients aged 55 or older are consistently more likely to have cardiovascular disease than not. Created and copyrighted by Ananya and Abhaya Saridena.

Fitting Model to Data

To fit a model to data, it's necessary to divide the dataset into two sections. The first section trains the model by feeding it $X$, which is composed of features from the dataset, and $Y$, containing target values; for instance, whether or not a patient has CVD. Once trained on $X$ and $Y$ in tandem, the model correlates specific features with presence- or absence of Cardiovascular disease likelihood. The second section-- called validation set-- evaluates how well the model performs.

Before splitting the training and validation sets into $X$ and $Y$, the data is normalized. The normalization of data refers to the process of setting a common scale between the various features. The process of normalization ultimately aids the model in identifying patterns effectively. The following code is used to normalize the dataset:

```python
max_val = df_train.max()
min_val = df_train.min()
df_train = (df_train - min_val) / (max_val - min_val)
df_valid = (df_valid - min_val) / (max_val - min_val)
```

In the excerpt above, the max_val and min_val variables store the maximum and minimum values of each feature in the training set. Then, the features in both the training and validation sets are scaled using these maximum and minimum values so that they are all on the same scale. This prevents any skewed results due to inconsistencies in data. Next, the training and validation sets are split into their respective feature sets (train_X and val_X) and target variables (train_y and val_y). The feature sets contain all the columns except the target variable, while the target variables contain only the values indicating whether a patient has cardiovascular disease (CVD).

Developing the Model

Once the data is processed, the subsequent stage entails creating a deep learning model by defining its architecture. The model comprises distinct layers, each serving specific functions and contributing to the decision-making process of the model. This particular deep learning prototype was established utilizing TensorFlow Keras library—a well-known open-source machine learning platform renowned for its beginner-friendly features. Additionally, Keras is an artificial neural network using TensorFlow as its backend processor. Below code accurately outlines the underlying structure of this AI system:

```python
model = keras.Sequential(
    layers.Dense(units=100, activation='relu', input_shape=[train_X.shape[1]],)
    layers.Dropout(rate=0.2),
    layers.Dense(units=100, activation='relu'),
    layers.Dropout(rate=0.2),
    layers.Dense(units=1, activation='sigmoid')
)
```

The Keras Sequential API allows developers to create deep learning models in a step-by-step fashion, adding one layer at a time for easy architecture definition. In this code example, the model begins with a fully connected dense layer containing 100 neurons. The Rectified Linear Unit (ReLU) is used as the activation function for non-linearity and capturing complex data patterns. Setting input_shape parameter ensures the correct number of
The shape of the model is based on the number of features in train_X, which helps prevent overfitting by employing dropout layers. Dropout randomly turns off a certain percentage (20% in this model) of the nodes during training to ensure more robust learning and less reliance on specific features or overfitting. The final layer has one unit with a sigmoid function, squashing output between 0 and 1, indicating a patient's probability of having cardiovascular disease as per the model's prediction.

Compiling the Model
Once the model is defined, the next step is to compile it. Compilation involves specifying the optimizer, loss function, and metrics used to evaluate the model's performance. The following code accomplishes this:

```python
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

The 'adam' optimizer is utilized in this model. It's an algorithm that tweaks a neural network during training to minimize the loss function and improve performance by finding the best parameter values, leading to more precise predictions. The Adam optimizer stands out in deep learning as it combines AdaGrad and RMSprop algorithms' merits for better results. This technique adjusts each parameter's learning rate based on past gradients behavior, ensuring quicker convergence and more stable outcomes, mainly when dealing with rare gradients scenarios. Therefore, implementing the 'adam' optimizer was crucial for these reasons. It can expertly handle complex and high-dimensional parameter spaces. 'Adam' comes with the added benefit of being able to adapt learning rates, which results in a smoother and more-efficient navigation process. This prevents issues such as vanishing or exploding gradients, ultimately leading to superior convergence and better overall model performance.

Early Stopping
After the model has been put together, the next step is to define early stopping. Early stopping is a method used to avoid overfitting and to get through the optimal number of epochs during the training of a machine learning model. If performance on the validation set does not increase after a predetermined number of epochs, the training process is discontinued. This entails tracking validation loss or other defined parameters during training. To do this, the following code was implemented:

```python
early_stopping = EarlyStopping(patience=20, restore_best_weights=True)
```

This excerpt shows the implementation of the EarlyStopping callback, which applies the EarlyStopping function. The callback's patience parameter was set to 20 epochs, meaning that if no improvement in validation loss is observed for 20 consecutive iterations, the training process will end. This enables avoiding overfitting to the training data by halting further model updates when further refinement appears unproductive after a certain point. In machine learning and deep learning, "weights" refer to the parameters that are learned during the training process. These weights are the coefficients assigned to the input features of a model and determine the strength or importance of each feature in making predictions. The restore_best_weights parameter is set to True, ensuring that the weights of the model are restored to the configuration that yielded the best validation accuracy. This is used to retrieve the model with the best performance on the validation set, even if the training process is stopped early due to lack of improvement. During each epoch until EarlyStopping triggers, pandas tracks and visualizes the validation loss and accuracy providing insight into the model's performance (figures 8, 9).
Training the Model

Finally, the model is trained by iterating through the training data for a specified number of loops. During this training process, the observed loss is used to regulate and update the model’s weights. To begin this process, use the following code to train:

```python
history = model.fit(
    train_X, train_y,
    validation_data=(val_X, val_y),
    batch_size=64,
    epochs=1000,
    callbacks=[early_stopping],
    verbose=1
)
```

The model object uses the fit method in order to train the model. The training features (train_X) and target variables (train_y) are inputted. To provide the validation data, including validation features (val_X) and target variables (val_y), use the validation_data parameter. The batch_size is set to 64 with each update of weight occurring after processing 64 data points at a time. The epochs parameter determines the number of times the model will cycle through the entire training dataset. The early_stopping callback defined earlier is passed to the callbacks parameter, allowing the model to stop early if the validation loss does not improve. The verbose parameter is set to 1 to display progress updates during training. The training process generates a history object, which contains information about the model's performance during training, such as the loss and accuracy at each epoch.

Results

To test the accuracy of the model, multiple trials were conducted. In each trial, the data was split differently and the model was run on that data, outputting graphs of the validation loss, training loss, val_loss, and loss. The following table contains the results of these trials. It is important to note that these were 10 random trials and doing another ten trials will entail similar, but not identical results to the trials below.
Figures 8 and 9. A graph of validation loss and training loss, and a graph of training accuracy and validation accuracy. Values represented are from trial 3 as referenced in table 2. Created and copyrighted by Ananya and Abhaya Saridena.

Table 2. Validation accuracy varies from trial to trial, along with validation loss. Created and copyrighted by Ananya Saridena

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Val. Accuracy</th>
<th>Val. Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7296</td>
<td>0.5541</td>
</tr>
<tr>
<td>2</td>
<td>0.7283</td>
<td>0.5534</td>
</tr>
<tr>
<td>3</td>
<td>0.7263</td>
<td>0.5527</td>
</tr>
<tr>
<td>4</td>
<td>0.7287</td>
<td>0.5543</td>
</tr>
<tr>
<td>5</td>
<td>0.7295</td>
<td>0.5556</td>
</tr>
<tr>
<td>6</td>
<td>0.7278</td>
<td>0.5580</td>
</tr>
<tr>
<td>7</td>
<td>0.7266</td>
<td>0.5564</td>
</tr>
<tr>
<td>8</td>
<td>0.7264</td>
<td>0.5589</td>
</tr>
<tr>
<td>9</td>
<td>0.7255</td>
<td>0.5601</td>
</tr>
<tr>
<td>10</td>
<td>0.7259</td>
<td>0.5611</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7278</td>
<td>0.55646</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.00454</td>
<td>0.002859</td>
</tr>
</tbody>
</table>

The final model had an average accuracy of 72.78% with a highest accuracy of 72.96%. The rest of the tests resulted in accuracies between 72.55% and 72.95%. Thus, the creation of a simple deep learning prediction model using readily available resources is not only simple, but also easily accessible and implementable, allowing for an improved detection method of CVD. It is important to note that extensive research into CVD is not necessary for the production of an artificial intelligence model.
Conclusion

Cardiovascular disease stands to be a major healthcare burden as the leading cause of death globally. This study proposes a deep learning model to alleviate this burden through the detection of CVD. The paper explains key information regarding cardiovascular disease including risk factors, specific types, and current diagnostic techniques. Meanwhile, it also explains the steps required to create a simple, accessible, and fast deep learning model using publicly available datasets. Currently, CVD diagnosis is mostly done using clinical methods that can require expensive laboratory tests to be performed regularly. Accurate and accessible supervised learning models can help speed up the diagnostic process, allowing for more time to be spent on successfully treating patients. The proposed model uses multiple layers, activation functions, optimizers, and loss functions, detecting cardiovascular disease with up to 73% accuracy. The lowest percentage accuracy is 72.55%. In suggesting a guide towards creating a simple deep learning program, this paper can shape future research in understanding the use of machine learning models in the medical field in preventing other similar future diseases.

Acknowledgements

I would like to thank my mentor Mr. Rajagopal Appavu and my advisor Coach Jo for all of the support and guidance you have given me along the way.

References


