Student’s Name

Department, Institutional Affiliation

Course Name and Number

Professor’s Name

Due Date

An Introduction to Deep Learning Using MATLAB

Deep Learning

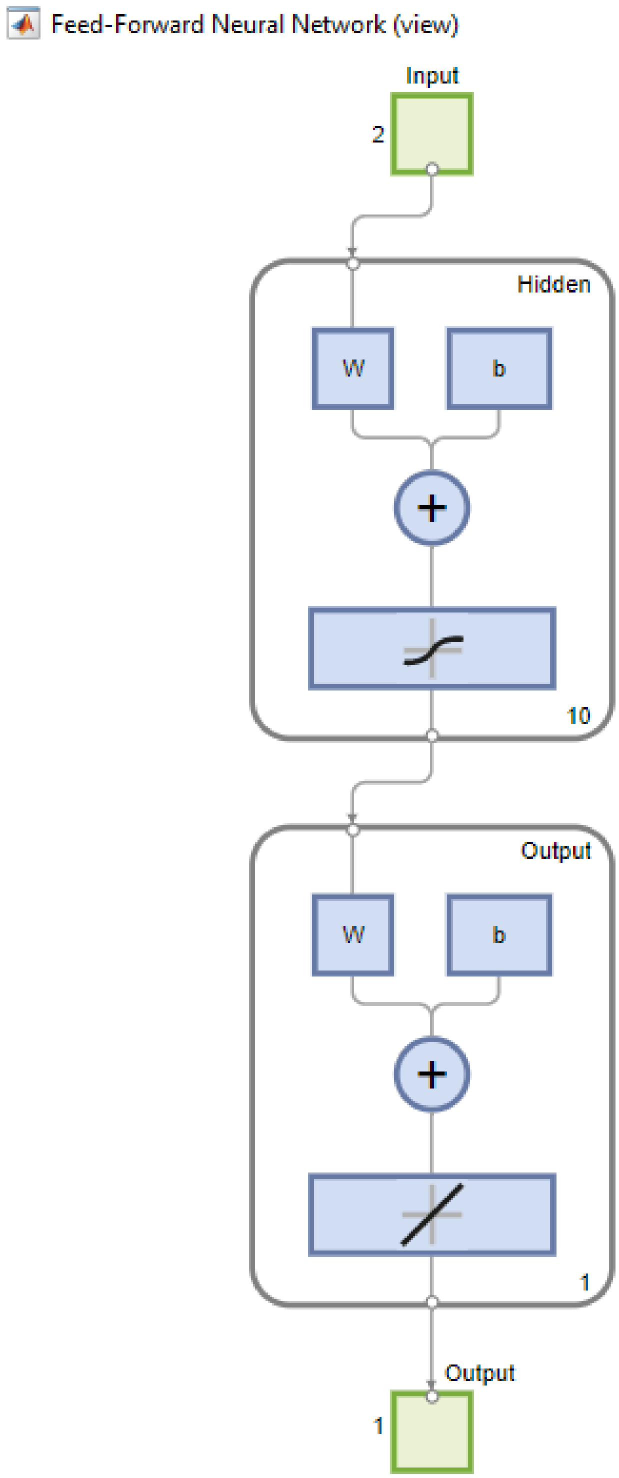
A branch of machine learning called "deep learning" is devoted to teaching artificial neural networks how to carry out different tasks (Bengio.et al 2017). In this MATLAB-based project, we apply a feedforward neural network to a basic binary classification problem and introduce deep learning using a synthetic dataset.

Feedforward Neural Network

We learned the decision boundary between two classes in a synthetic dataset using a feedforward neural network, which is a basic kind of neural network (Chen. et al. 2022). With ten neurons in its hidden layer, this network has sufficient capabilities to record the underlying link between features and labels.

Figure 1:

Neural Network Model

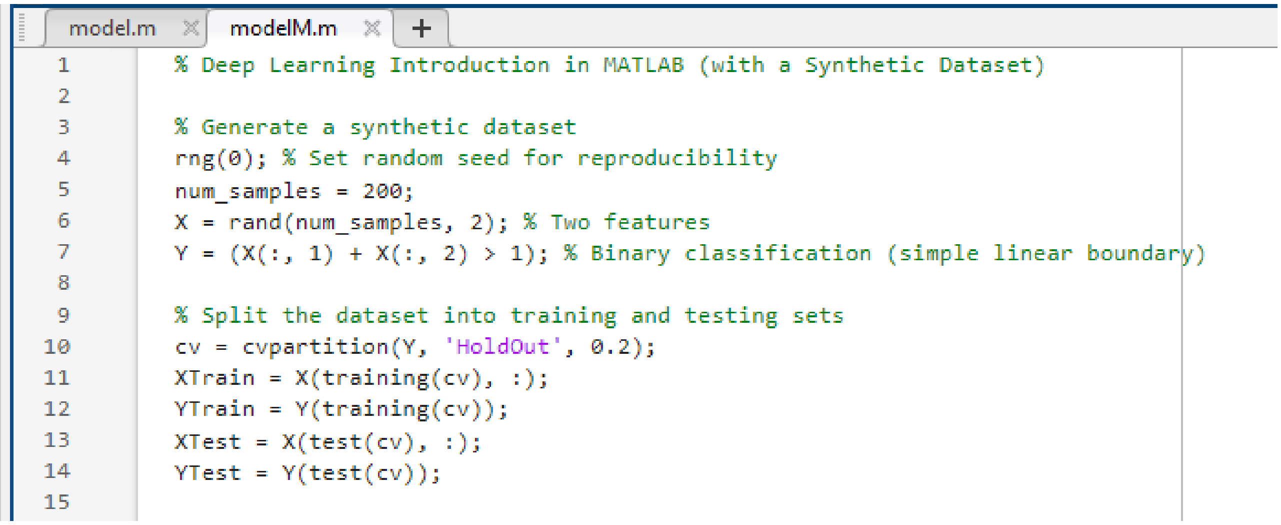


Data Preparation

To begin, we generate a synthetic dataset with 200 samples, each having two features. We set a random seed for reproducibility. The dataset is then split into training and testing sets with a 20% holdout for testing.

Figure 2:

Screenshot showing Data Generation and Splitting



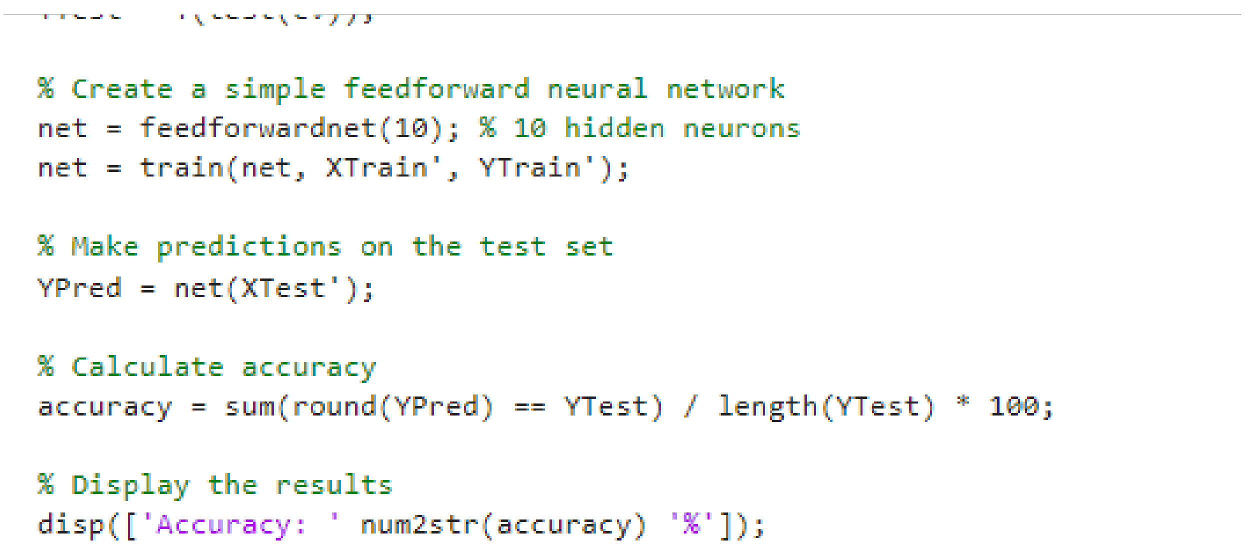
```matlab
% Deep Learning Introduction in MATLAB (with a Synthetic Dataset)

% Generate a synthetic dataset
rng(0); % Set random seed for reproducibility
num_samples = 200;
X = rand(num_samples, 2); % Two features
Y = (X(:, 1) + X(:, 2) > 1); % Binary classification (simple linear boundary)

% Split the dataset into training and testing sets
cv = cvpartition(Y, 'HoldOut', 0.2);
XTrain = X(training(cv), :);
YTrain = Y(training(cv));
XTest = X(test(cv), :);
YTest = Y(test(cv));

```

Model Building/Implementation

Using the training data, we build a feedforward neural network and train it. Based on the given features, the network is intended to learn the decision boundary that separates the two classes.

Figure 3:

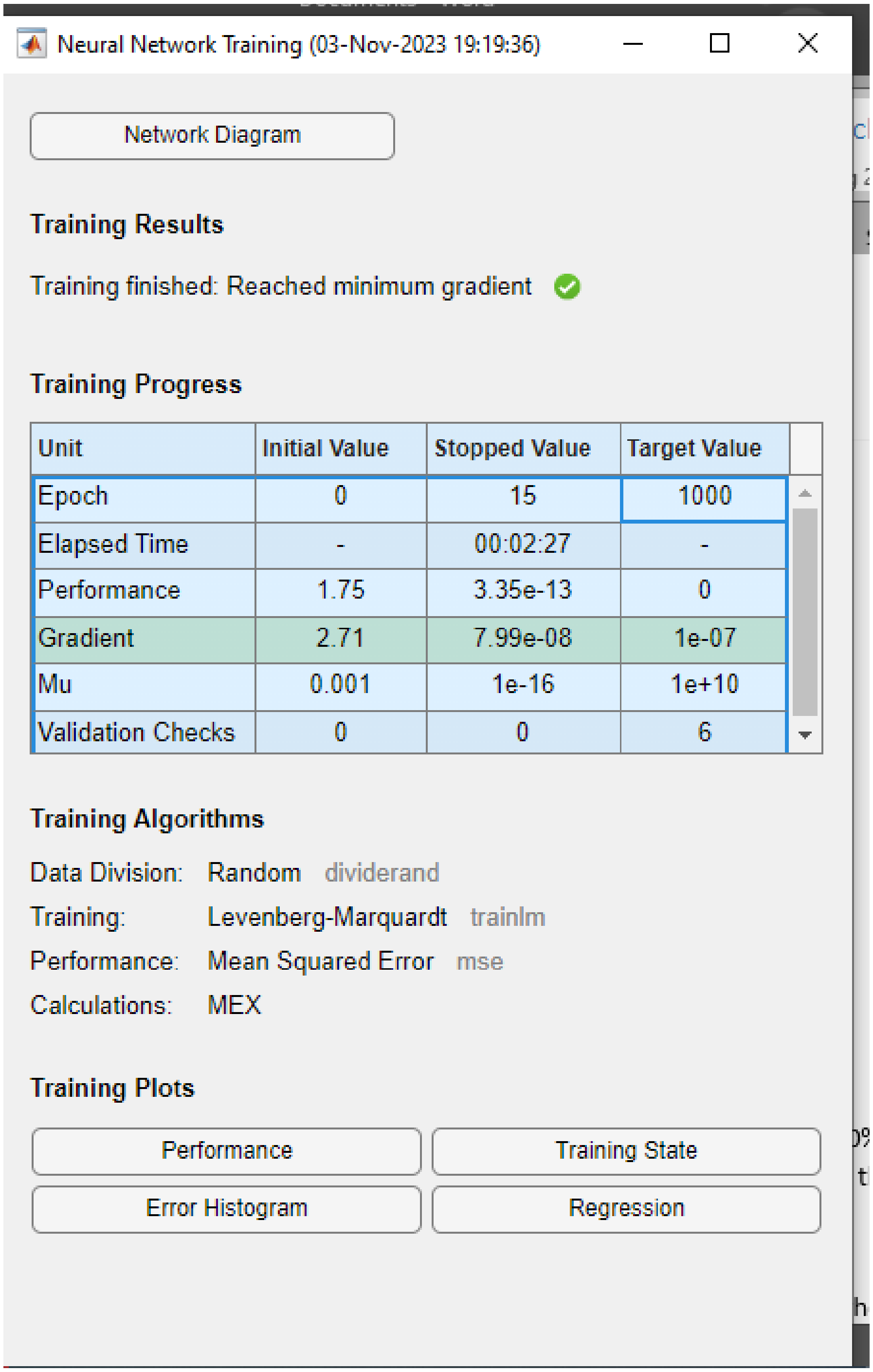
Screenshot showing the Neural Network Implementation in Matlab



```
% Create a simple feedforward neural network
net = feedforwardnet(10); % 10 hidden neurons
net = train(net, XTrain', YTrain');

% Make predictions on the test set
YPred = net(XTest');

% Calculate accuracy
accuracy = sum(round(YPred) == YTest) / length(YTest) * 100;

% Display the results
disp(['Accuracy: ' num2str(accuracy) '%']);
```

Figure 4:

Screenshot showing the Training Report



Neural Network Training (03-Nov-2023 19:19:36)

Training Results

Training finished: Reached minimum gradient ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	15	1000
Elapsed Time	-	00:02:27	-
Performance	1.75	3.35e-13	0
Gradient	2.71	7.99e-08	1e-07
Mu	0.001	1e-16	1e+10
Validation Checks	0	0	6

Training Algorithms

Data Division: Random dividerand

Training: Levenberg-Marquardt trainlm

Performance: Mean Squared Error mse

Calculations: MEX

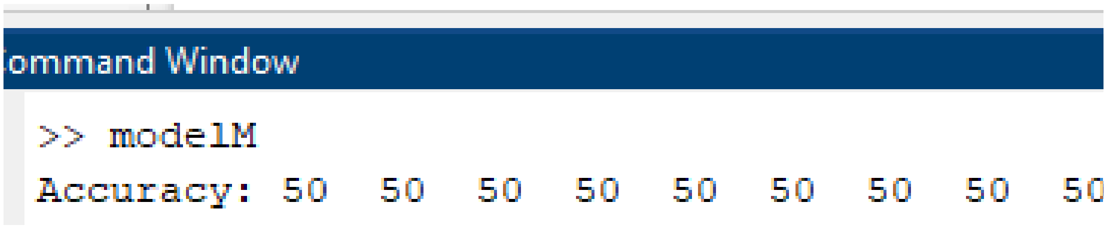
Training Plots

Results and Discussion

The accuracy of the model trained on the test dataset is calculated and found to be 50%. This result indicates that the model is not effectively learning the underlying relationship between the features and the labels.

Figure 5:

Screenshot showing the Accuracy Value obtained

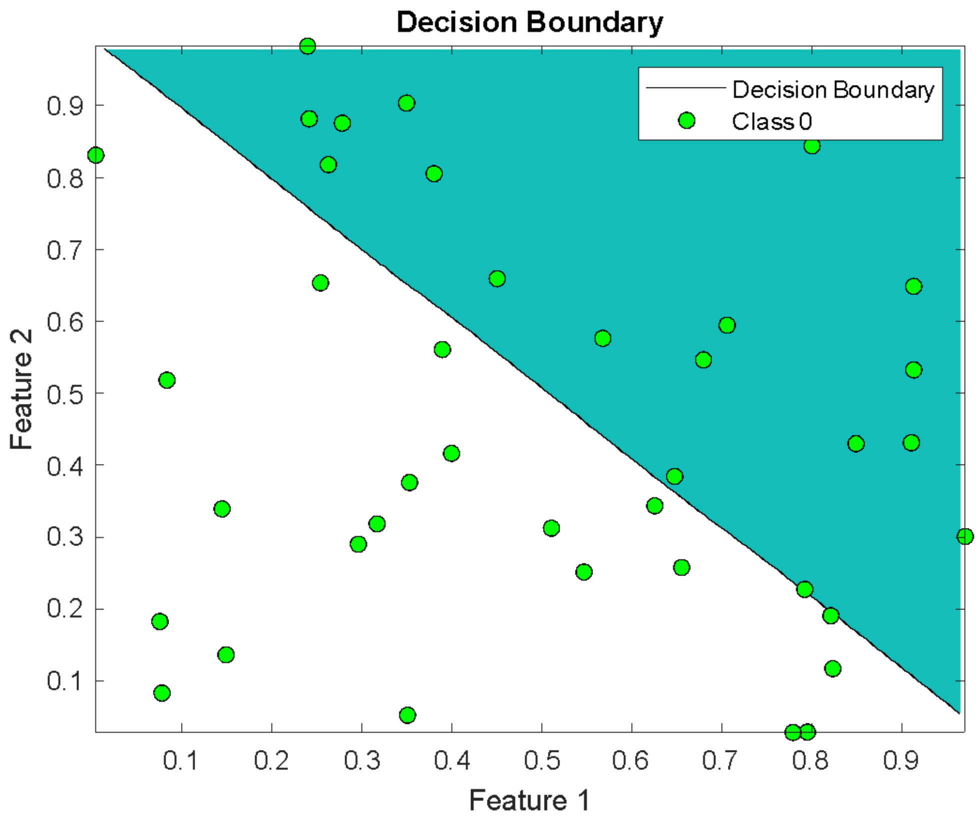


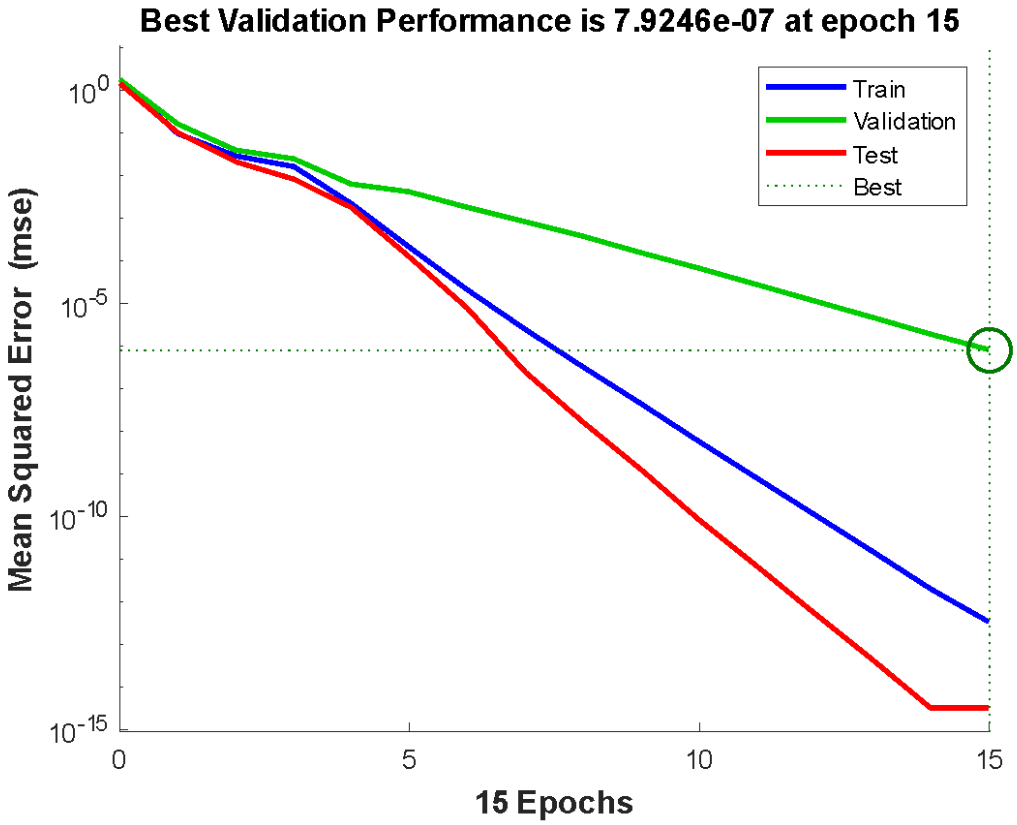
```
>> modelM
Accuracy: 50   50   50   50   50   50   50   50   50
```

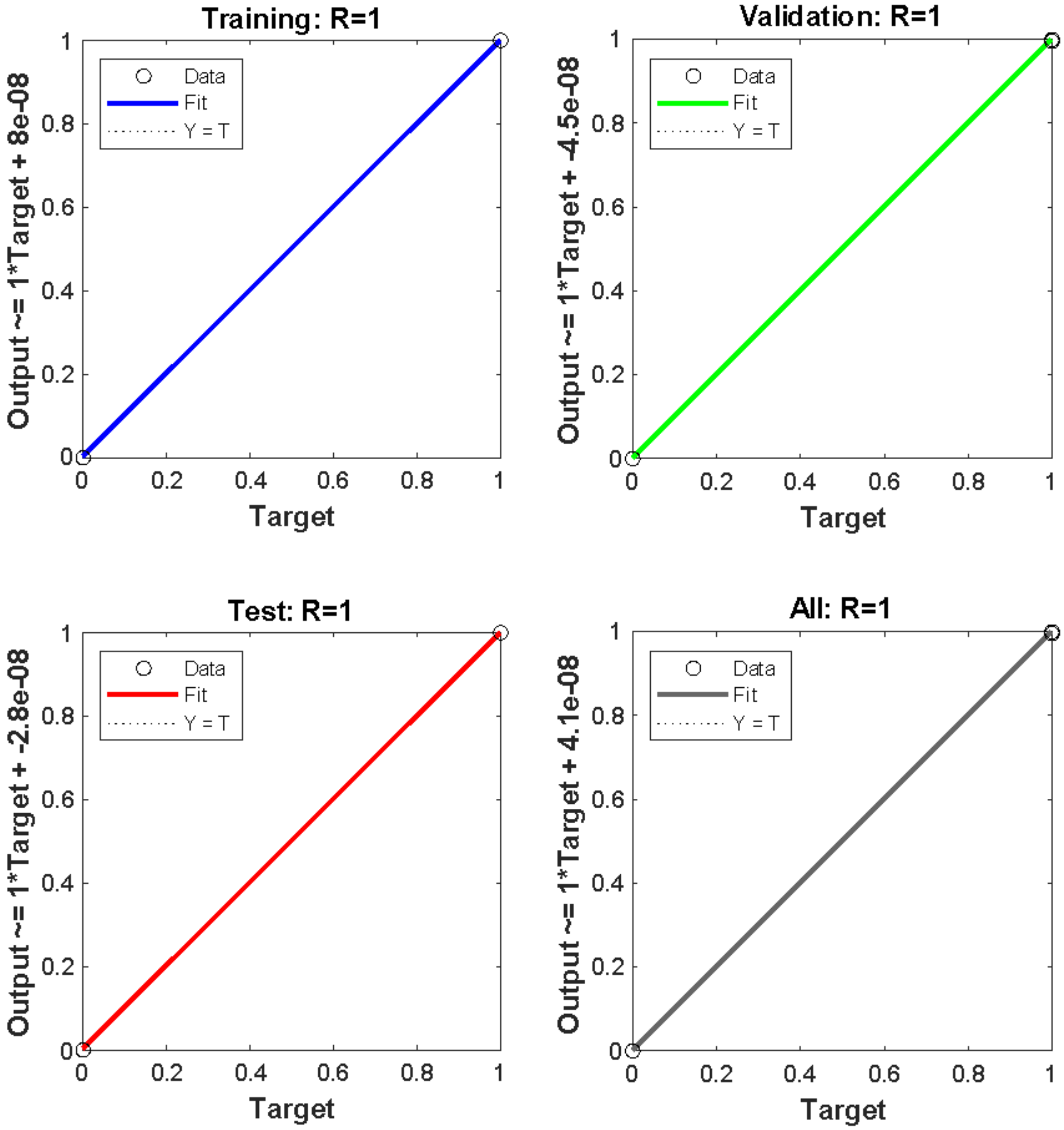
The decision boundary is also visualized using a contourplot.

Figure 6:

Decision Boundary Plot




Best Validation Performance is 7.9246e-07 at epoch 15
Mean Squared Error (mse)
10^{0}
10^{-5}
10^{-10}
10^{-15}
0
5
10
15
15 Epochs
Train
Validation
Test
Best



Conclusion

In this introductory exercise, we used MATLAB to implement a basic deep learning model. Although the model shows the fundamental stages of data preparation, model construction, and evaluation, the problem's simplicity and the model's architecture limit the model's performance. To attain superior outcomes in intricate, real-world situations, deep learning models with suitable architectures are essential.

References

Bengio, Y., Goodfellow, I., & Courville, A. (2017). Deep learning (Vol. 1). Cambridge, MA, USA: MIT press.

Chen, Y., Zhang, C., Liu, C., Wang, Y., & Wan, X. (2022). Atrial fibrillation detection using a feedforward neural network. Journal of Medical and Biological Engineering, 42(1), 63-73.