The Bioscan AN INTERNATIONAL QUARTERLY JOURNAL OF LIFE SCIENCES 19(2): S.I (1), 294-301, 2024 www.thebioscan.com

Prediction of Heart Disease by Developing the Hybrid Deep Learning Models to Attain Higher Accuracy

Mukta Sandhu [1], Dr M Jithender Reddy [2], Dr. M. Bheemalingaiah [3], Dr. KARTHIKEYAN C [4], Dr. A. Azhagu Jaisudhan Pazhani [5], Dr Subba Rao Polamuri [6]

1 Shri Vishwakarma Skill University, Palwal, India. Mukta.sandhu@gmail.com

2 Department of Computer Science and Engineering, Vasavi College of Engineering, Hyderabad, India. m.jitenderreddy@staff.vce.ac.in

3 Department of Computer science and engineering, J.B. Institute of Engineering and Technology. Hyderabad, India. bheemasiva2019@gmail.com

4 Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil nadu-602105.
Mail ID: ckarthik2k@gmail.com

5 Department of Electronics & Communication Engineering, Ramco Institute of Technology, Rajapalayam, Tamilnadu,
Mail: alagujaisudhan@gmail.com

6 Department of Computer Science and Engineering, Aditya University, Surampalem, India. psr.subbu546@gmail.com

DOI: https://doi.org/10.63001/tbs.2024.v19.i02.S.I(1).pp294-301

KEYWORDS

CNN,
Deep Learning,
LSTM,
ML,
Data analysis.

Received on:

08-08-2024

Accepted on:

25-11-2024

ABSTRACT

One of the most common long-term diseases in the world is cardiovascular disease, also called heart disease. It is hard to make accurate and quick predictions about heart problems. Most of the work that has been done so far to identify heart disease has used machine learning methods, but they have not been able to get more accurate results. Recent advances in deep learning methods have a big effect on data analysis. Combining convolutional neural networks via a memory (LSTM), which network is what this work is all about. It aims to be additional exact than added ML methods. The heart disease information was put into two groups: normal and abnormal. This was done using a mixed CNN and LSTM method. The k-fold cross-validating method remained used to prove that this combination system works 90% of the time. Different machine learning algorithms, like SVM, Naïve Bayes, and Decision Tree, are compared with the suggested method to see how well it works. The outcomes show that future method works improved than the ML models that are already in use.

INTRODUCTION

Heart disease, also known as cardiovascular disease (CVD), is one of the main reasons why the death rate is going up around the world [1]. Over 12 million individuals die every year from heart disease, according to the WHO. Any problem with the heart can be called a heart disease. Heart diseases include infections, genetic flaws, blood tube diseases, heart valve diseases, and more [2]. This makes it harder to pump or circulate the right number of bloods to all parts of the body. Heart diseases can be caused by many things, like smoking, high blood pressure, cholesterol levels, diabetes, being overweight, stress, and so on. A lot of people have had chest pain and tiredness, but some didn't know they had them until they happened. So, it's important to keep an eye on the signs that lead to CVD [3]. Angiography is the most common way to find out what's wrong with the heart. But it's very expensive and you need to know a lot about technology to do it [4-6].

Besides that, different tests are used to find out if someone has heart disease, such as an ultrasound, an electrocardiograph, electrical studies, cardiac perfusion scans, and a tilt table test [7-12]. But skilled and experienced doctors are needed to find out if someone has heart disease. Cardiovascular disease has been studied in the past in a number of different ways. The main way that heart disease is diagnosed is with machine learning techniques. Because they can pull out the important data from a lot of data, machine learning methods are useful [13-15]. A decision tree, naive Bayes, a support vector machine, the radial source operation, K-NN, and single conjunctive rule classifier remain some of ML methods that been floated in past. It was only 66–86% accurate with these methods, though [16-20].

So that the sickness isn't misclassified and is found early, an automatic method is needed to help find it before it gets worse. The main tool used to identify chronic diseases was artificial intelligence [21]. An increasing number of experts are interested in deep learning algorithms these days since they are good at handling complicated tasks [22]. With its help, the model can learn more complicated tasks. It instantly learns the traits of the given data.

In medical uses, different deep learning methods are used to handle difficult jobs. CNN, proved to be good way towards identify heart disease after looking at different algorithms [23]. This is because as the levels get deeper, the model is easier to learn, & the structures can be shown in short, general way. The long-term & short-term memory models is likewise put forward in this effort, along with CNN [24]. A time-varying recurrent neural networks is what LSTM is. The LSTM is made up of parts that can talk to each other and store and send back information as needed [25,26]. This is why the CNN+LSTM combination model is suggested, which makes the classification more accurate.

2. Foreground:

2.1. Convolutional Neural Network:

One well-known method in ANN is the CNN. It is a type of deep learning that is often used in picture processing. This is recognized way to get features, and it can be utilised in the sorting process. It usually has a fully linked layer, a pooling layer, and a convolutional layer. Each stimulus has a weight and an opinion that go with it. Figure 1 shows the standard CNN design.

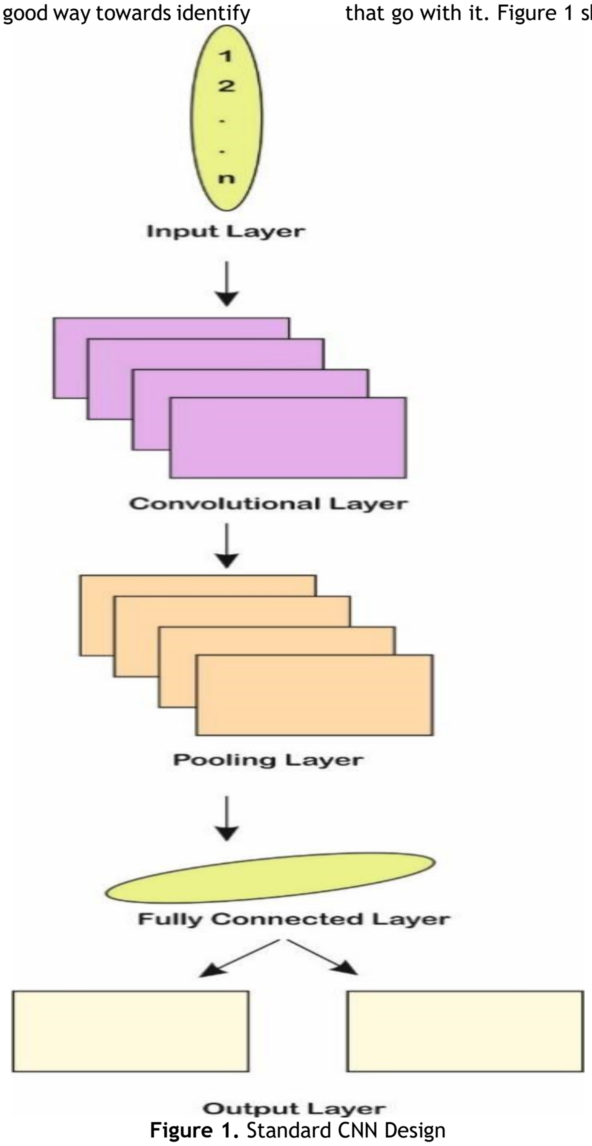

Figure 1. Standard CNN Design

2.1.1. Convolutional Layer:

In order to generate the feature map, the convolution of the input information using weights is computed in this layer, added to the bias, and then sent into the activation function. The next layer receives this feature map. The output may be computed using Eq. (1) if the value of the input is x_j.

$$o_j = f(b_j + \sum_{j=1}^{p} w_j x_j) \qquad (1)$$

where w_j is the layer's weight, f is an activation function, x_j has the input vectors, o_j is the output, and b_j is the bias.

2.1.2. Pooling Layer:

The pooling layer comes after the convolutional layer and does pooling processes to make the features that the convolutional layer calculated smaller. Max pooling along with average pooling are two common ways to pool. Maximum sharing is used in this suggested work, and it is a good method. With the given inputs, this process finds the highest number that can be reached. Equation (2) shows the max-pooling processes.

$$f_j = \max\,(o_j) \qquad (2)$$

Feature values are turned through a single vector along with sent to the fully linked layer after going through several layers of convolution and pooling.

2.1.3. Layer with Full Connection:

The layer that is completely connected gets that which comes out of the sharing layer. The result from the pooling layer is sent to a fully linked layer, which guesses which trait is most likely to match the class name. A SoftMax algorithm was used as the activation purpose in this effort. There is tool called SoftMax classifier that turns the data into a probability distribution. This classifier does a good job with binary classifiers.

2.2. LSTM:

A kind of RNN that is utilised a lot in deep learning is the LSTM method. Most LSTM units have one cell, an input gates, an output gates, along with a fail to recall gate. The cell can tell the difference between numbers at random times, as well as the three leftover gates control the flow of information from and to the cell.

The standard LSTM design is shown in Figure 2. A single-layer neural network manages memory in the forgot gate. It uses sigmoid beginning to figure out the output by using the input, the output from the preceding block, & memory. Equation (3) shows the purpose.

$$fn_j = h\left(w\left[x_j, s_{j-1}, m_j - 1\right] + b_{fn}\right) \qquad (3)$$

It has a sigmoid function (h), an input vector (x), a previous unit output (s), a previous gate memory (m), weights (w), and a bias (b). As you can see in Equations (4) and (5), the neural network made original memory in the input gate while utilising the tanh purpose via the memory from previous gate.

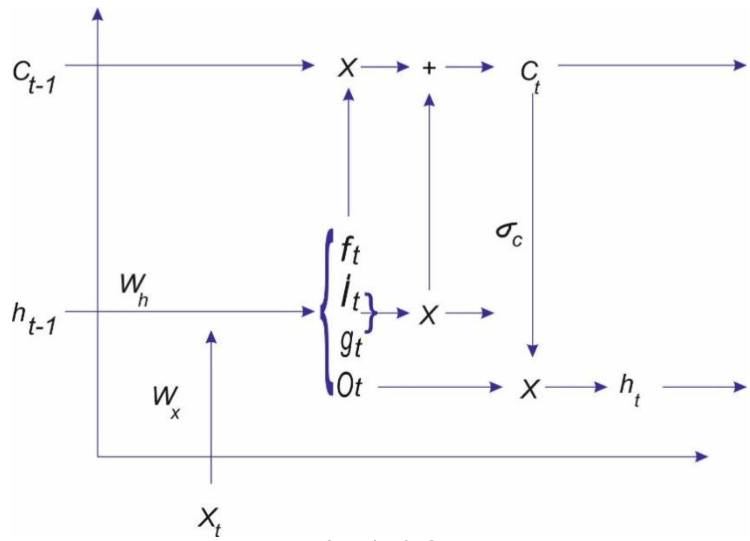

Figure 2. Standard LSTM Design

$$ip_j = h\left(w\left[x_j, s_{j-1}, m_{j-1}\right] + b_{ip}\right) \qquad (4)$$

$$C_j = fn_j \times C_{j-1} + ip_j \times \tanh\left(x_j, s_{j-1}, m_{j-1}\right] + b_c \qquad (5)$$

Finally, outputs are made in the output gate. Equations (6) and (7) can be used to figure this out.

$$Op_j = h\left(w\left[x_j, s_{j-1}, m_{j-1}\right] + b_{op}\right) \qquad (6)$$

$$g_j = op_j \times \tanh\left(C_j\right) \qquad (7)$$

3. Proposed Mission:

3.1. Dataset:

The UCI ML library dataset is used in this work. Table 1 shows this dataset. The collection is made up of data collected by Hungarian Organization for Cardiology and Cleveland Hospital. There is information about both regular and odd medical records in it. There are 304 observations in this database, which is made up of 77 characteristics. The characteristics are things like age, gender, blood pressure at rest, cholesterol, and so on. There are also six unknown numbers in the data set. Out of 304 observations, 139 were of normal people and 166 were of abnormal people, meaning they had heart disease.

3.2. Pre-processing:

Once the data has been gathered, it needs to be prepared to deal with noise and missing numbers. There are six unknown numbers in this set of data. Care should be taken with the missing numbers because they could change the results. There are a number of preprocessing methods that can be used, including cleaning, integrating, reduction, transformation, along with discretisation. The normalisation method is utilised in this work to deal with the missing numbers in the data set. To change the numbers on a shared scale, normalisation methods are used. Z numbers are used to fill in the blanks. The set of collected data is normalised using z-scores in Equation (8).

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \qquad (8)$$

while x is an input vector and max(x) and min(x) are the dataset's highest and lowest values, respectively. Once the data have been cleaned up, they are split into two groups: sets for training and testing. The data for this project are split into two groups: training data (71%), and testing data (31%).

3.3. Selection of Features:

Information Acquisition, Gini Index, PCA, along with weight by SVM are all good ranking methods for choosing features. When you use the information gain along with Gini index, you check the gain as well as Gini index for each characteristic and choose the one with the highest gain and Gini index. The information is split into groups for sorting based on these characteristics. An orthogonal transformation is used by Principal Component Analysis (PCA) to change linked qualities into uncorrelated attributes. This paper looks at weight through SVM feature selection methods. To get an idea of how much each trait is worth, it produces the F-score as shown in Equation (9).

$$FS(i) = \frac{\left(\overset{-(+)}{a_i} - \overset{-}{a_i}\right)^2 - \left(\overset{-(-)}{a_i} - \overset{-}{a_i}\right)^2}{\frac{1}{n_+ - 1}\sum_{i=1}^{n+}\left(\overset{-(+)}{a_{i.j}} - \overset{-(+)}{a_i}\right)^2 + \frac{1}{n_- - 1}\sum^{n-}\overset{}{a_{i.j}} - \overset{-(-)}{a_i}\Big)^2} \qquad (9)$$

The input vector is denoted by a_i, and n_+ and n_- are the numbers of positive and negative training instances. The average of these instances is shown by $\overset{-}{a_i}$. The regular of both positive & negative cases is shown by $\overset{-(+)}{a_i}$ and $\overset{-(-)}{a_i}$. The ith characteristic of the jth positive along with negative case is shown by $\overset{-(+)}{a_{i.j}}$ and $\overset{-(-)}{a_{i.j}}$.

3.4. Hybrid of LSTM and CNN:

The pre-processed data is sent to convolutional layer, & then to the layers of pooling in this mixed network. The LSTM gets the data from the pooling layer, & entirely linked layer will guess if an individual is heart illness or not. Figure 3 shows the suggested design. This mixed network is made up of 5 layers that switch between convolutional & pooling layers. It is possible to teach the network structure to make the classifier work better and stop the parameters from fitting too well.

The pre-processed data that was received is then sent through the convolutional layer. The incoming grid is jumbled by this layer, which then sends it to the layer for pooling. A max-pooling action is done within layer of pooling, & result sent to layer of LSTM. The completely linked layer is given the tanh function by the LSTM layer. It is the fully linked layer's activation function (SoftMax classifier) that does the work, and the output layer gets the result.

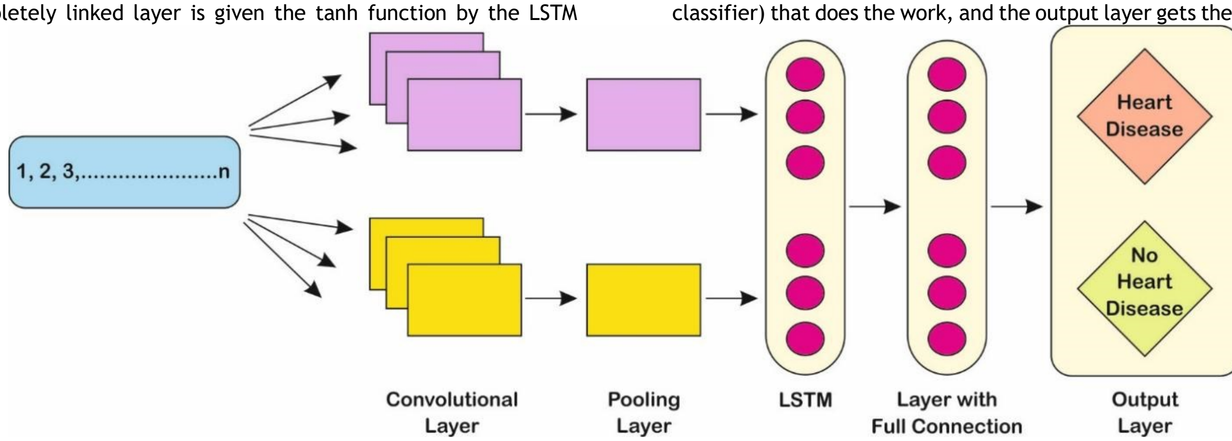

Figure 3. Possible Architecture Combining CNN and LSTM

Based on the outcomes from the entire layer of connection, the production layer sorts the information into two groups: heart illness and no heart illness. The backpropagation network is used to train the network. The gradient descendent function is used by the backpropagation method to find the fault. The weights & bias remain then changed based on the error number. The mistake is sent back to the layers that came before it, and then the neural network is trained it. Following the steps in Equations (10) and (11), the biases and weights are changed.

$$\Delta W_1(p+1) = -\frac{x_\lambda}{r}\frac{x}{n}W_1 - \frac{\partial C}{\partial W_1} + m\Delta W_1(p) \tag{10}$$

$$\Delta B_1(p+1) = \frac{x}{n}\frac{\partial C}{\partial B_1} + m\Delta B_1(p) \tag{11}$$

l is the number of layers, n is entire amount of training the examples, m is momentum, & C is cost function. W is the weight, B is the bias, λ is the regularisation limitation, x is the learning rate, along with p is update stage. In all of convolutional layers, (Rectified Linear Units) ReLu are utilised as an event trigger. It is in the output layer that the SoftMax activation function is used. A batch normalisation layer was added to this network to lower the overflowing and compassion of the algorithm. It likewise helped with the backpropagation purpose.

4. Outcomes of the Experiment:

This part looks at how well mixed CNN and LSTM can predict heart disease. Multiple tests were done on mixed LSTM and CNN networks to find the best settings for deep learning. The pseudocode is shown under.

Aimed at the Hybrid CNN-LSTM Algorithm, here is a few fake code:

```
Input: UCI Dataset
Output: Predict the persons normal having no heart Disease(0) and abnormal who
are having heart disease(1)
Procedure CNN_LSTM model (Train_data, Test_data)
model=Sequential()
#Initialization
batchsize=",No_of_epochs=",filters=",Poolsize,verbose="
#Convolutional Layer
modeladd(Conv1D(kernel size, activation_function=")
#Pooling layer
model.add(MaxPooling(Poolsize)
#LSTM layer
model.add(LSTM)
#Fully Connected Layer
model.add(Dense(classes, activation=")
#compile function
model.Compile(loss=gradient_descendant_function, optimizer=
'Adam')
#Fitting a model
for all epochs in {1to N} do
model.fit (Train_data, Validation (Test_data, batch_size=")
#model evaluation
model_evaluation(Test_data, verbose, batch_size")
Predict Accuracy, Precision and Recall score
End for
Return Acc score
End Procedure
```

Normalising the inputs makes sure that they all fall between 0 and 1. The convolutional layer is given the features that were chosen. This layer has two parameters: (1) kernel size, which tells it how many dimensions the convolutional window has; & (2) beginning function, which in this case is ReLu. The convolutional layer's name is assigned to layer that pools. Max pooling is used on all of the data to get the features that are needed from the information by picking the highest numbers. Researcher use the Sigmoid activation function to figure out the odds for the two groups: heart disease and no heart disease. This is done in the layer that is completely connected. The model is set up in the compile purpose after getting the outcomes of the fully linked layer. The limitations utilised are (1) a loss purpose to lower the damage in the method and (2) an optimiser to build model.

As part of building a model, the test results are used to judge the model. In the end, accuracies are found, which show how well the algorithm worked with the test data. CNN+LSTM is run 201 times. ADAM is employed as an optimiser, and it learns at a rate of 0.002. Dual cross-entropy is the loss function that is used. Measures of accuracy, precision, and sensitivity are used to judge how well the model works. These numbers show how good and stable the model is. Specificity checks how well the actual denials are predicted, while sensitivity checks how effectively the true positives are predicted. Equations (12), (13), and (14), which find the success measures, are used.

$$ACC = \frac{TPr + TNr}{TNr + FPr + TPr + FNr} \times 100 \tag{12}$$

$$SPEC = \frac{TNr}{TNr + FPr} \times 100 \quad (13)$$

$$SENS = \frac{TPr}{TPr + FNr} \times 100 \quad (14)$$

It shows that *TPr* stands for positive information that are accurately estimated as normal, *TNr* for adverse results that are properly forecasted as irregular, *FPr* for negative information that has been forecast as usual, & *FNr* for optimistic information that are forecast as irregular. Receiver operation characteristics are used to find out the diagnosis quality. In binary classification problems, the ROC curve is a common way to measure how well a model is doing. Here is a graph that shows the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) for all levels. The ROC curve to the suggested model can be seen in Figure 4. Thus, it shows that the model is good at telling the difference between good and bad classes.

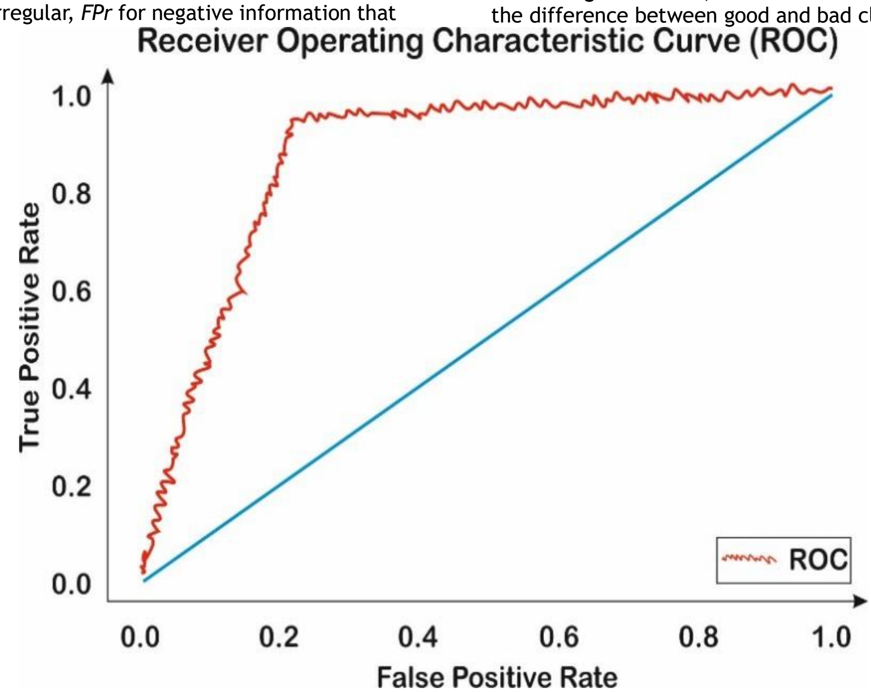

Figure 4. ROC Curve

There is hyperparameter called "epoch" that describes one filled run finished the whole training dataset. Additionally, it mean the total number of times a program runs until it gets better. Using each example in the collection, it trains the network. Figure 5 shows the Accuracies Vs Epoch. The outcome showed that the network got very good at both training and testing data whenever the epoch number went up. In 201 epochs, CNN+LSTM networks got results that were 90% accurate.

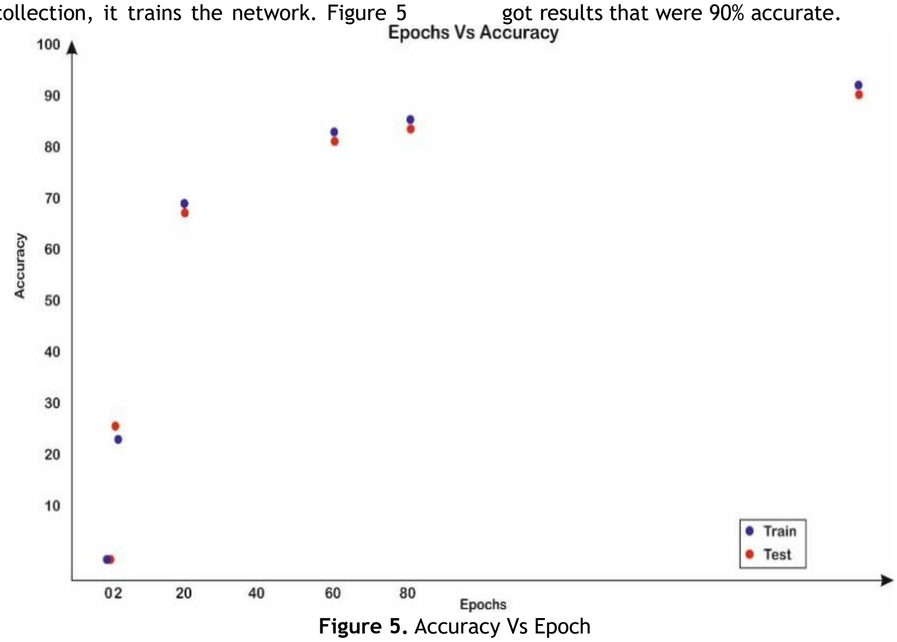

Figure 5. Accuracy Vs Epoch

Figure 6 shows how a network is handled at different learning rates. The accuracy changed as the number of rounds and learning rates went up. CNN+LSTM worked very well at a learning rate of 0.001.

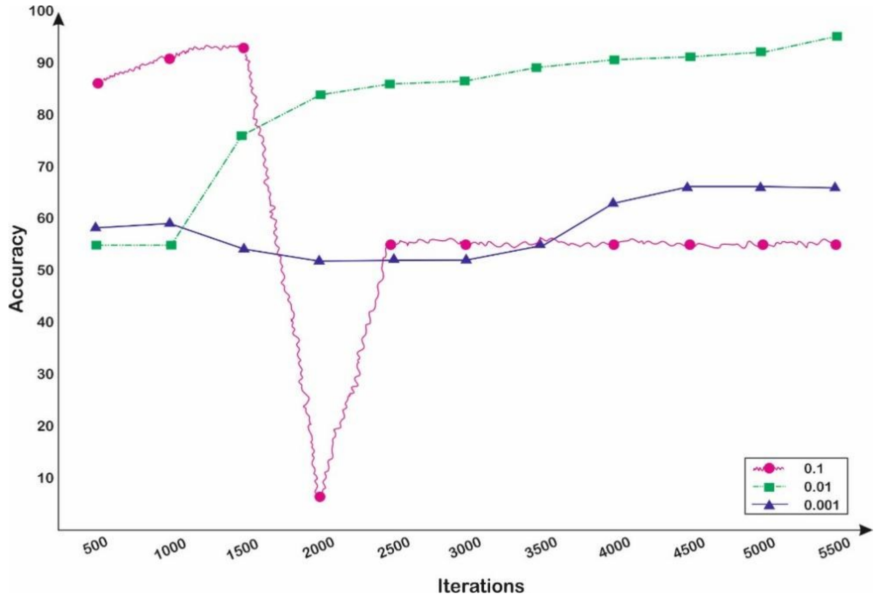

Figure 6. Accuracy Vs Learning rate

As shown in Figure 7, when CNN works on its own, both accuracy and loss went up at the same time for higher CNN levels. But when it was paired with LSTM, the decline went down, and the accuracy went up. It has also been seen that precision and accuracy are the same thing. Higher accuracy was reached by adding more secret layers and making a combination of LSTM and CNN. Several machine learning algorithms were compared to the suggested method. Figure 8 shows that CNN+LSTM was the most accurate of these. Because, unlike RNN and LSTM, CNN doesn't employ the information it already has to process. LSTM sorts future instances into groups based on what it knows about past instances. So, CNN via LSTM is used, and it does a good job of handling the changing nature of the big amount of data.

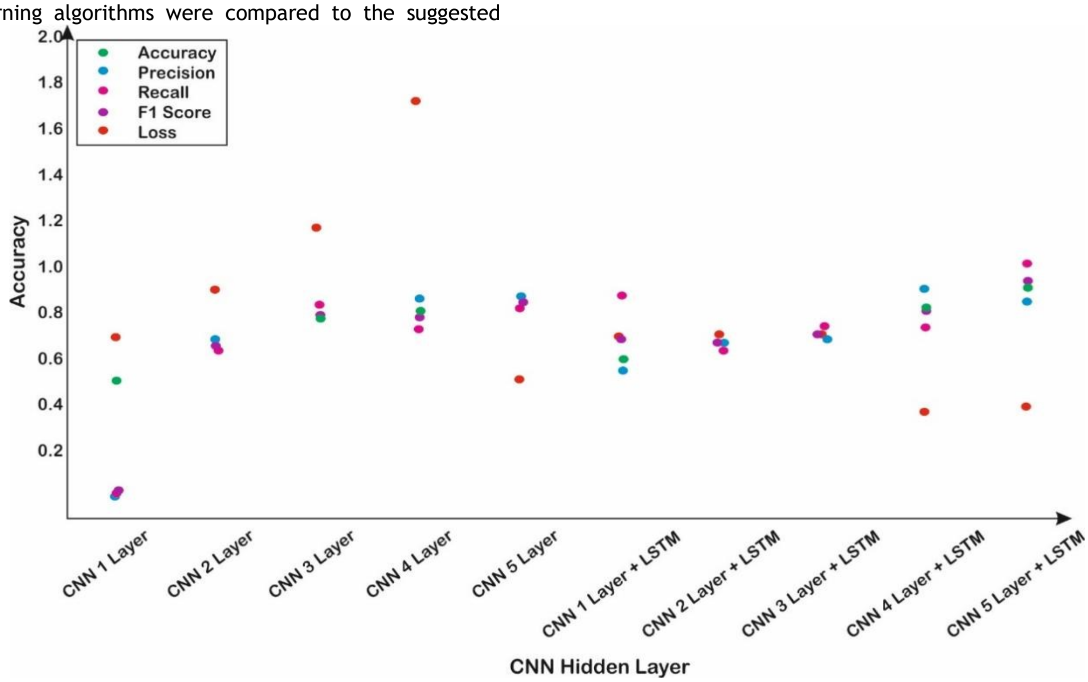

Figure 7. CNN and LSTM layers Comparison

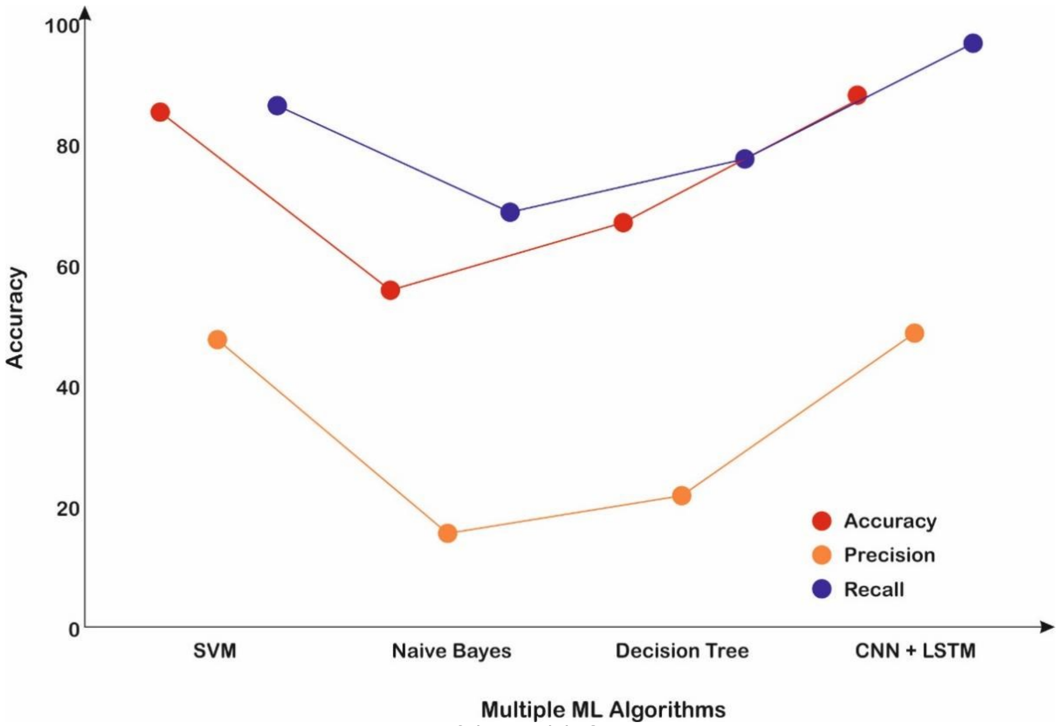

Figure 8. Other Models Comparisons

CONCLUSION

Heart disease is a big problem for most people around the world. The sickness can't be fixed, but it may be kept under control. People with heart disease must be found as quickly as possible because if it is not managed, it can kill them. A good way to classify heart diseases is suggested in this paper by combining deep learning methods like the CNN (Convolutional Neural Network) along with Long Short-Term Memory (the LSTM). Normalisation is used to deal with missing numbers before the data are sent to the computer. For choosing features, the weights by SVM technique is utilised, & the characteristics are assumed to a CNN & an LSTM models that works together. The technique of backpropagation is utilised as well to keep the model's loss as low as possible. Comparisons of different factors are used to study the proposed model, and each time the suggested hybrid model does better. Compared to other standard machine learning models, the results of the experiments showed an accuracy of 90%, 82% sensitivity, along with 94% specificity. It will be possible to test this CNN+LSTM model with real-time medical information in the future and look at how well it works.

REFERENCES

- Kavitha, M., et al. "Heart disease prediction using hybrid machine learning model." *2021 6th international conference on inventive computation technologies (ICICT)*. IEEE, 2021.
- Mohan, Senthilkumar, Chandrasegar Thirumalai, and Gautam Srivastava. "Effective heart disease prediction using hybrid machine learning techniques." *IEEE access* 7 (2019): 81542-81554.
- Ramesh, B., and Kuruva Lakshmanna. "A Novel Early Detection and Prevention of Coronary Heart Disease Framework Using Hybrid Deep Learning Model and Neural Fuzzy Inference System." *IEEE Access* 12 (2024): 26683-26695.
- Ali, Farman, et al. "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion." *Information Fusion* 63 (2020): 208-222.
- Dwarakanath, B., et al. "A novel feature selection with hybrid deep learning based heart disease detection and classification in the e-healthcare environment." *Computational Intelligence and Neuroscience* 2022 (2022).
- Bharti, Rohit, et al. "Prediction of heart disease using a combination of machine learning and deep learning." *Computational intelligence and neuroscience* 2021.1 (2021): 8387680.
- Sk, Khader Basha, et al. "Coronary Heart Disease Prediction and Classification using Hybrid Machine Learning Algorithms." *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*. IEEE, 2023.
- Saboor, Abdul, et al. "A method for improving prediction of human heart disease using machine learning algorithms." *Mobile Information Systems* 2022.1 (2022): 1410169.
- Naseer, Arslan, et al. "An improved hybrid model for cardiovascular disease detection using machine learning in IoT." *Expert Systems* (2023): e13520.
- Kahramanli, Humar, and Novruz Allahverdi. "Design of a hybrid system for the diabetes and heart diseases." *Expert systems with applications* 35.1-2 (2008): 82-89.
- Fitriyani, Norma Latif, et al. "HDPM: an effective heart disease prediction model for a clinical decision support system." *IEEE Access* 8 (2020): 133034-133050.
- Taj, Shaik Ghouhar, and K. Kalaivani. "Hybrid Prediction Model with Improved Score Level Fusion for Heart Disease Diagnosis." *Computational Biology and Chemistry* (2024): 108278.
- Rani, Pooja, et al. "A decision support system for heart disease prediction based upon machine learning." *Journal of Reliable Intelligent Environments* 7.3 (2021): 263-275.
- Kumar, Mikkili Dileep, and K. V. Ramana. "Cardiovascular disease prognosis and severity analysis using hybrid heuristic methods." *Multimedia Tools and Applications* 80.5 (2021): 7939-7965.
- Mehmood, Awais, et al. "Prediction of heart disease using deep convolutional neural networks." *Arabian Journal for Science and Engineering* 46.4 (2021): 3409-3422.
- Dutta, Aniruddha, et al. "An efficient convolutional neural network for coronary heart disease

prediction." *Expert Systems with Applications* 159 (2020): 113408.
- Kuveskar, Manisha R., et al. "A hybrid iot-enabled transfer learning model for accurate diagnosis of coronary heart disease." *Journal of Electrical Systems* 20.2s (2024): 1145-1158.
- BramahHazela, et al. "Machine Learning: Supervised Algorithms to Determine the Defect in High-Precision Foundry Operation." Journal of Nanomaterials 2022.1 (2022): 1732441.
- Gaikar Vilas Bhau, Radhika Gautamkumar Deshmukh, T. Rajasanthosh kumar, Subhadip Chowdhury, Y. Sesharao, Yermek Abilmazhinov, IoT based solar energy monitoring system, Materials Today: Proceedings, Volume 80, Part 3, 2023, Pages 3697-3701, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.07.364
- El-Hasnony, Ibrahim M., et al. "Multi-label active learning-based machine learning model for heart disease prediction." *Sensors* 22.3 (2022): 1184.
- Cocianu, Cătălina-Lucia, et al. "Classical, evolutionary, and deep learning approaches of automated heart disease prediction: a case study." *Electronics* 12.7 (2023): 1663.
- Diwakar, Manoj, et al. "Latest trends on heart disease prediction using machine learning and image fusion." *Materials today: proceedings* 37 (2021): 3213-3218.
- Bakar, Wan Aezwani Wan Abu, et al. "A review: heart disease prediction in machine learning & deep learning." *2023 19th IEEE international colloquium on signal processing & its applications (CSPA)*. IEEE, 2023.
- Arabasadi, Zeinab, et al. "Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm." *Computer methods and programs in biomedicine* 141 (2017): 19-26.
- Verma, Luxmi, Sangeet Srivastava, and P. C. Negi. "A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data." *Journal of medical systems* 40 (2016): 1-7.
- Sivaprasad, R., et al. "Heart Disease Prediction and Classification using Machine Learning and Transfer Learning Model." *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*. IEEE, 2022.