A Hybrid Bidirectional LSTM and 1D CNN for Heart Disease Prediction

Mohamed G. El-Shafiey[1], Ahmed Hagag[2,*], El-Sayed A. El-Dahshan[3], and Manal A. Ismail[4],

[1]Faculty of Computers and Information Technology, Egyptian E-Learning University, Dokki, Giza, 12611, Egypt.
[2]Faculty of Computers and Artificial Intelligence, Benha University, Benha, 13518, Egypt.
[3]Department of Physics, Faculty of Science, Ain Shams University, Abbasia, Cairo 11566, Egypt.
[4]Faculty of Engineering, Helwan University, Helwan, Cairo, 11731, Egypt.

Summary
One of the essential parts of human life is healthcare. Heart disease is one of the worst diseases, claiming the lives of millions of people all over the world. Accordingly, heart disease prediction is considered a significant aspect of clinical data analysis. Therefore, numerous research has been conducted to create machine-learning algorithms for the early detection of heart disorders in order to assist clinicians in the design of medical procedures. Traditional methods have been limited by their inability to generalize adequately to new data not seen in the training set. This paper proposes a hybrid bidirectional LSTM and 1D CNN architecture with Bayesian optimization for hyperparameters to increase the accuracy of heart disease prediction. The performance of the proposed 1D CNN-BiLSTM approach is validated via evaluation metrics, namely, accuracy, specificity, sensitivity, and area under the receiver operating characteristic (ROC) curve by using two datasets from the University of California, namely, Cleveland and Statlog. The experimental results confirm that the proposed approach attained the high heart-disease-prediction accuracies of 89.01% and 82.72% on the Cleveland and Statlog datasets, respectively. Furthermore, the proposed approach outperformed other state-of-the-art prediction methods.

Key words:
Cleveland dataset; 1D CNN; Bi-LSTM; Heart-disease prediction; Statlog dataset.

1. Introduction

Heart disease is the leading cause of mortality in today's world. The constriction of coronary arteries that feed blood to the heart is the most common cause of heart disease, also known as cardiac disorders. There are several ways for identifying heart problems, such as Angiography. However, they are pretty expensive and might cause specific responses in the body of the patient. This keeps these approaches from being widely used in nations having a significant number of impoverished people.

According to the latest World Health Organization (WHO) data, heart disease is responsible for approximately 37% of all fatalities worldwide. According to the predictions, heart disease will raise the global death rate by 2030 [1]. Therefore early identification of heart disorders is important for reducing Heart Failure (HF) symptoms and extending patients' lives [2]. Many researches have recently been conducted in order to enhance the early detection of heart problems and minimize fatalities. Machine learning and deep learning have been utilized in several research articles to diagnose heart disease and predict if a patient has heart disease. Helwan et al. [3] presented a backpropagation neural network (BPNN) to identify heart disease utilizing six hidden layers and a 60:40 data sharing ratio. This demonstrates that BPNN is more effective and precise in the detection of heart disease, with an accuracy rate of 85%. Deep learning is a modern artificial intelligence approach that has been used for successful analysis in a variety of disciplines. Deep learning approaches speed up processing large amounts of data effectively, outperforming standard machine learning algorithms like SVM, random forest, and Naive Bayes. Although studies have been conducted by many authors in heart disease prediction, implementing hybrid deep learning methods and investigating more in the pre-processing step is still insufficiently explored to improve the accuracy rate of heart disease prediction.

The main contribution of this paper is to develop a hybrid approach, called 1D CNN Bi-LSTM, for heart-disease prediction. First, data pre-processing is applied. After that, we implemented a hybrid bidirectional LSTM and 1D CNN architecture with Bayesian optimization for hyperparameters. Finally, the proposed 1D CNN Bi-LSTM is validated via evaluation metrics, namely, accuracy, specificity, and area under the receiver operating characteristic (ROC) curve by using two heart-disease datasets from the University of California (UCI), Irvine, machine learning repository [4], namely, Cleveland and Statlog. Experimental results show that the proposed method achieves high prediction accuracies.

The rest of this paper is structured as follows. Section 2 illustrates the related work. The materials and proposed approach are discussed in Section 3, including the description of both the datasets, describe the architecture of the proposed method, and classification process. The experimental results are provided in Section 4, including a comparative analysis of our method with those in the literature. Finally, the conclusions are drawn in Section 5.

Manuscript received October 5, 2021
Manuscript revised October 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.10.18

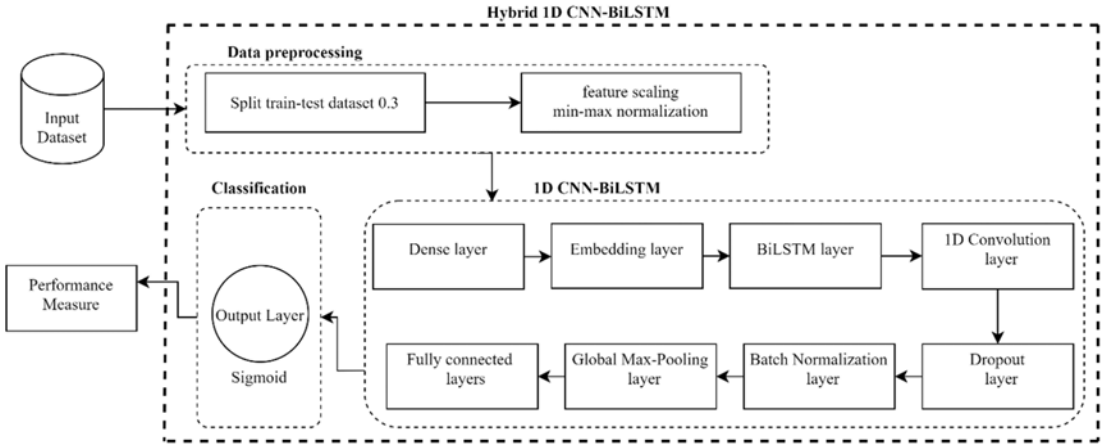

Fig. 1. Block Diagram of the proposed method.

2. Related Works

Several publications have appeared in recent years focus on developing a heart disease diagnosis system using machine learning, deep learning, and data mining techniques for early prediction. The recently published related researches are summarized in this section.

Subbalakshmi et al. [5] implemented the Nave Bayes data mining approach to predict heart disease from 303 cases (UCI Machine Earning Repository), with an accuracy of 82.31%. Sundar et al. [6] applied Nave Bayes to identify the probability of a patient developing the heart disease, based on [5]. A weighted associative classifier and Nave Bayes obtained recognition rates of 84% and 78%, respectively. The data mining approach is also employed by Chaurasia et al. [7] to examine heart disease utilizing 11 features from the UCI Machine Learning Repository. They applied Nave Bayes and J48 decision trees to model the data, with recognition rates of 82.31% and 84.35%, respectively. Revett et al. [8] deployed the use of rough sets to determine the information content of each subset of the feature space. Subsequently, on the Cleveland, Hungarian, Switzerland, and SPECTF datasets, Saqlain et al. [9] used the Fisher score and Matthews correlation coefficient as an FS method and SVM for binary classification to identify heart disorders. Three performance measures were used to verify their method: accuracy, specificity, and sensitivity. Because the authors utilized FS in their research, they needed to evaluate a variety of FS algorithms to improve prediction accuracy.

Recently, Mohammed et al. [10] developed a hybrid model by combining ANN, SVM, and NB for cardiac disease prediction. They achieved an overall accuracy of 88%. After that, Chu-Hsing et al. [11] proposed employing deep learning models to identify heart illness; the author utilized the Cleveland dataset. In this study, a deep learning algorithm with varying numbers of hidden layers was evaluated. The model was also examined with and without a category model. The findings show that the accuracy attained is high. Ghosh et al. [12] used machine learning algorithms with Relief and least absolute shrinkage and selection operator (LASSO) feature selection techniques, such as gradient tree boosting [13].

3. Materials and Proposed Method

We aim to discriminate between patients with heart disease and those who are healthy. Therefore, the experimental method was developed as a hybrid bidirectional LSTM, and 1D CNN architecture with Bayesian optimization for hyperparameters is proposed to classify heart disease. The features were tested to evaluate how successful different machine learning (ML) techniques were at diagnosing heart disease. Validity and efficiency evaluations metrics for the model were computed. We perform a data pre-processing step, in which an embedding layer is implemented on categorical features to be converted to embedding vectors. The overall workflow of the proposed method is illustrated in Fig. 1. Four tasks have to be performed for heart disease prediction: (1) data pre-processing, (2) 1D CNN-BiLSTM utilization, (3) binary classification, and (4) performance measurement. In the following subsections, we describe the datasets, and then each step of the proposed method is discussed [4].

3.1 Datasets Description

Two datasets from the UCI machine-learning library, Statlog, and Cleveland are employed in the proposed method [4]. The (Num) variable represents two heart disease diagnosis values: 0 signifies healthy (the patient has no heart disease), and 1 indicates unhealthy (the patient has the heart disease). In the Statlog dataset, 120 records have the value (1), while 150 have the value (0), as seen in Fig.

2. In addition, 165 records in the Cleveland dataset have the value (1), whereas 138 have the value (0). The dataset is divided into two parts. We used 70% for training, and the remaining 30% is used for the test set (unseen data). The training set is further split into a train set and validation set to train the algorithm and optimize the parameters.

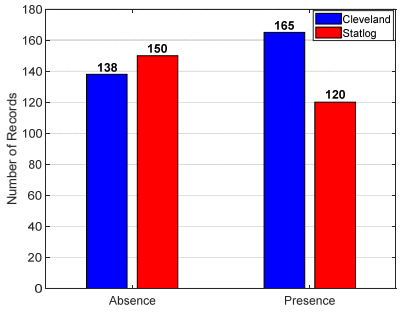

Fig. 2. Distributions for the Statlog and Cleveland datasets.

3.2 Data pre-processing

The data of numerical features are scaled using min-max normalization. We implemented fit transform on the training set and transformed it for validation and test set. This process has been gaining importance because all features may have different data types. It eliminates the numerical difficulties due to the different range of values during the computation process. This technique converts a value a to $\grave{a}$ in the range of $[max\_new - min\_new]$ as follows:

$$\grave{a} = \frac{a - a_{min}}{a_{max} - a_{min}} \times [max\_new - min\_new] + min\_new; \quad (1)$$

where from min_new to max_new denotes the range of the transformed values. We implemented $min\_new = 0$ and $max\_new = 1$. After that, these transformed values were used as input for the 1D CNN-LSTM architecture.

3.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning technique that uses convolutional neural networks. It is a type of feed-forward artificial neural network that is widely utilized [14]. CNN was created by LeCun in the early 1990s [15]. A CNN is a multilayer perceptron that is comparable to a multilayer perceptron (MLP). The design of the model permits CNN to exhibit translational and rotational invariance because of this unique structure [16]. In general, a CNN consists of one or more convolutional layers, related weights and pooling layers, and a fully connected layer [17]. The convolutional layer uses the local correlation of the information to extract features.

3.3.1 Convolutional layer

This layer computes a dot product (or convolution) of each subregion of the input data using a kernel, adds it with a bias, and then passes it through an activation function to generate a feature map for the following layer [18, 19].

If the input vector for beat samples is $x_i^0 = [x_1, x_2, \ldots, x_n]$, and n is the number of samples per beat; then, the output values are calculated using Eq. (2) [19].

$$C_i^{l,j} = h(b_j + \sum_{m=1}^{M} w_m^j x_{i+m-1}^{0j}) \quad (2)$$

Hither, l is the layer index; h denotes the activation function used to impart non-linearity to this layer, and b denotes the bias term for the j^{th} feature map. The kernel/filter size is specified by M, while the weight for the j^{th} feature map and m^{th} filter index is specified by w_m^j.

3.3.2 Batch normalization

Batch-by-batch, the training data is acquired. Consequently, the batch distributions are non-uniform and unstable and must be fitted using the network parameters in each training cycle, significantly slowing the model's convergence. To address this issue, a convolutional layer is followed by an adaptive reparameterization technique known as batch normalization. The batch normalization method determines the mean μ_D and variance σ_D^2 of each batch of training data and then adjusts and scales the original data to zero-mean and unity-variance. Furthermore, weight and bias are applied to the shifted data $\hat{x}_l$ to enhance their expressive power. Equations 3-6 provide the necessary computations. The batch normalization algorithm's reparameterization greatly relieves coordinating updates across layers in the neural network.

$$\mu_D = \frac{1}{m} \sum_{i=1}^{m} x_i \quad (3)$$

$$\sigma_D^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_D)^2 \quad (4)$$

$$\hat{x}_l = \frac{x_i - \mu_D}{\sqrt{\mu_D^2 + \epsilon}} \quad (5)$$

$$y_i = \gamma \hat{x}_l + \beta \quad (6)$$

3.3.3 Max-pooling layer

The pooling layer is also known as the sub-sampling layer. In the proposed method, the 1-D max-pooling layer is applied after the 1-D convolutional layer and batch normalization layer, which conducts a down sampling operation on the features to decrease their size [18]. It

accepts small rectangular data chunks and generates a unique output for each block [18]. This can be accomplished in a variety of ways. The Maxpooling procedure is employed in this paper to determine the maximum value in a collection of adjacent inputs [19]. Eq. (7) defines the pooling of a feature map within a layer [19].

$$p_i^{l,j} = \max_{r \in R} c_{i \times T+r}^{l,j} \quad (7)$$

The pooling window size is denoted by R, while the pooling stride is denoted by T. Subsequently, the acquired features are transformed to a single one-dimensional vector for classification using multiple convolutional and max-pooling layers [18]. These classification layers are fully connected, besides each classification label corresponding to a single output type. Compared to other techniques, such as depth, feedforward neural networks, CNN requires fewer experimental parameter values and minimum pre-processing and pre-training algorithms [20]. Consequently, it is a highly appealing framework for deep learning.

3.4 Bidirectional Long Short-Term Memory model

Since deep learning is the most sophisticated type of machine learning available today, there is a growing variety of neural network models available for application with real-world situations. In this paper, an effective deep learning method was utilized to demonstrate its distinctive and fascinating problem-solving techniques. It is referred to as the long short-term memory due to its memory-oriented features.

The Bi-LSTM is a deep learning technique that efficiently analyzes data and extracts the important features necessary for prediction. This technique is an extension of Recurrent Neural Network (RNN). The predecessors designed the new network structure of LSTM [21] to overcome the "vanishing gradient" problem of the previous RNN structure. Input gate, output gate, forgetting gate, and memory unit are all part of the LSTM structure (Cell) [22]. One-layer neural network controls the forget gate in the memory block structure. Eq. (8) [23] is used to determine the activation of this gate.

$$f_t = \sigma(W[x_t, h_{t-1}, C_{t-1}] + b_f) \quad (8)$$

Whereas the input sequence is denoted by x_t; h_{t-1} denotes the previous block output; C_{t-1} denotes the previous LSTM block memory; b_f denotes the bias vector. W denotes individual weight vectors for each input, σ while denotes the logistic sigmoid function.

The input gate is a part in which a basic NN with the tanh activation function and the prior memory block effect is used to create new memory. Eqs. (9) and (10) are used to compute these operations [23].

$$i_t = \sigma(W[x_t, h_{t-1}, C_{t-1}] + b_i) \quad (9)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh([x_t, h_{t-1}, C_{t-1}]) + b_c \quad (10)$$

Consciously designing and remembering long-term information may prevent long-term dependencies, which is the default behavior of LSTM in practice.

The one-way LSTM is reliant on past data; however, this is not always adequate. The Bi-LSTM analyzes data in two directions. The hidden layer of Bi-LSTM holds two values [24], one of which is used in the forward computation and the other in the reverse calculation. The final output of Bi-LSTM is determined by these two values, which tends to enhance the prediction performance [25].

3.5 1- Dimensional CNN-BiLSTM proposed method

The 1-dimensional CNN (1D CNN) is identical to traditional 2D CNN, except that the convolution operation is only performed to one dimension, resulting in a shallow architecture as shown in Fig. 3. in that can be readily trained on a regular CPU or even embedded development boards [26]. The convolution process aids in the discovery of meaningful hierarchical features from a dataset for classification. The following equation may be used to determine the dimensions of the output features after 1D CNN:

$$x = \frac{w + 2p - f}{s} + 1 \quad (11)$$

where x denotes the output dimension, and w denotes the size of the input features. The size of the filter used for convolutions is indicated by f, and p stands for padding, which is the addition of values to the border before performing convolution. The variable s means stride, which is the distance traveled after executing the convolution process.

The one-dimensional convolution operation is a linear process that cannot be used to classify nonlinear data. Most real-world datasets are nonlinear, necessitating nonlinear operations following convolution. An activation function is a type of nonlinear function. The most utilized activation functions include the sigmoid, hyperbolic tangent, rectified linear unit (ReLU), and Exponential Linear Unit (ELU). The proposed CNN architecture employs the ELU activation function, which is simple to implement and enables quicker processing.

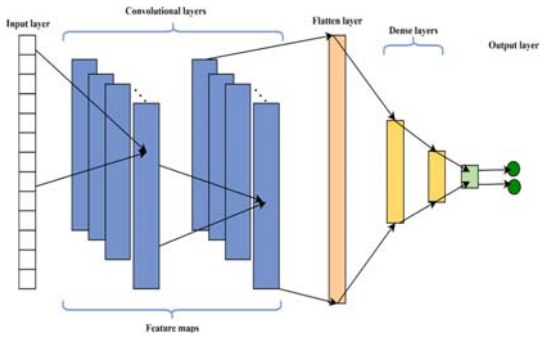

Fig. 3. The 1D CNN architecture.

Moreover, it fixes some of the problems with ReLUs and keeps some of the positive things. It also does not have any issues with disappearing or bursting gradients. The proposed 1D CNN-BiLSTM is illustrated in Algorithm 1.

3.5.1 Bayesian hyper-parameters optimization

Bayesian optimization is used to automatically adjust hyper-parameters to achieve a trade-off between data overfitting and the high accuracy of the proposed model. In the present study, the Bayesian optimization algorithm (BOA) determines the most suitable hyper-parameters. BOA is a very effective method for determining the extrema value of a black-box function, and it is beneficial when objective function evaluations are expensive [27]. The standard BOA [28] method works by fitting a Gaussian process model to the known data and calculating the posterior sample location using the Gaussian process model.

The best set of hyper-parameters of 1D-CNN extracted from the implementing Bayesian optimization is shown in Table 1. list of hyperparameters that includes num_dense_hidden_units which represent several dense layers, lstm_hidden_units denotes the dimensionality of the output space in LSTM, conv_filters represents several filters in 1D-convolution layer, out_dense_hidden_units represents the number of dense output layers, and emb_dimensions represents the dimensions for the embedding layer.

Table 1: Bayesian hyper-parameters optimization values for the proposed method.

Hyper-parameter	***Range of values***	***The best value***
num_dense_hidden units	From 32 to 512 (with step 32)	96
lstm_hidden_units	From 8 to 128 (with step 8)	16
conv_filters	From 8 to 128 (with step 8)	64
out_dense_hidden_ units	From 8 to 64 (with step 8)	48
emb_dimensions	From 30 to 300 (with step 30)	270

ALGORITHM 1: HYBRID PROPOSED METHOD 1D CNN-BILSTM

Input: Dataset D_{train} , D_{test} of data
CNN Parameters P
Output: Classification prediction results R of test data

1. Loading input data
2. Split the dataset into D_{train} and D_{test} - 70-30 split
3. Split D_{train} into train and validation set to train the algorithm and optimize the parameters
4. Data preprocessing of the dataset D_{train} , D_{test}
   a. Min-max normalization as shown in Eq. (1).
5. Initialize 1D CNN–BiLSTM model parameters
   a. As shown in Eq. (11)
6. Embedding layer to convert categorical features to embedding vectors
7. Dense layer to process the numerical features
8. BiLSTM layers enhance extracting the important features for prediction.
9. 1D CNN layer. Eq. (2)
10. Dropout layers to prevent overfitting
11. Batch normalization layer
    a. As described in Equations 3-6.
12. Global max pooling layer
    a. As shown in Eq. (7)
13. Dense layers
14. Fully connected layers
15. Sigmoid final layer
16. Get $P_{best} = getBestParams(P)$ using Bayesian hyper-parameters optimization
17. For $each\ train\ epoch$
18. Train the proposed model $Train(P_{best})$
19. End for
20. Evaluate the model on D_{train} , D_{test}
21. implement performance measures on classification

3.6 Classification

The input to the architecture will be the 13 features that are important in the classification of heart disease. The input to the architecture will be the 13 features that are important in the classification of heart disease. These features are converted to a new representation called word embedding by the layer called Embedding Layer. It is like the Bag of Words concept used for text data. This layer is implemented on categorical features to be converted to embedding vectors, which helps better represent the dataset according to unique values present in each of the features. The output of the Embedding layer is given to the 1D CNN layer for feature extraction. There can be multiple convolution layers in the architecture followed by an activation function. The proposed architecture uses two 1-D convolution layers with 128 filters and filter sizes of 1. The output of the final convolution layer is passed through the global max-pooling layer, which pools the maximum value from all the channels and reduces the dimension of output. The global max-pooling output is passed through the fully connected layer with 320 neurons which extracts the useful features for classification. To improve the performance of the CNN, the

convolutional and flatten layers are often followed by batch normalization and dropout layer. The CNN parameters were adjusted using the Adam optimizer [29], an adaptive learning-rate optimization method developed from the term "adaptive moments" to decrease the loss of the CNN. Adam calculates the gradients of the CNN parameters and updates them after each training cycle. The Adam optimizer was adjusted with a learning rate = 0.0002. The final layer contains a single neuron which gives the classification probability. The final layer uses the sigmoid activation function as it directly gives the probability for binary classification. The proposed 1D CNN-BiLSTM architecture contains around 0.09 million trainable parameters, which will get adapted during the network's training. It was observed that general CNN architecture overfitted the training data meaning that training accuracy was very high and validation accuracy was low. The dropout technique was introduced to remove overfitting. It removes random neurons with a certain probability during training which allows the difference.

3.7 Performance measures

Four metrics were used to evaluate the classification models' performance: accuracy, recall, precision, receiver operating characteristic (ROC), and area under the ROC curve (AUC). The rate of the correctness of a classifier is represented as accuracy. Consequently, we divide the total number of records by the sum of true positive (TP) and true negative (TN) records, which represents the sum of TN, TP, false negative (FN), and false-positive (FP). Thus, accuracy denotes the ratio of correctly predicted records to the total number of records, as shown in Eq. (12). The recall represents the rate of values that measures positive records that the classifier correctly predicted. Moreover, it is called true positive rate (TPR) or sensitivity. Thus, recall is calculated as shown in Eq. (13). Precision is the ratio of TP records to the total of positively predicted records, as shown in Eq. (14). The ROC curve is a graph of TPR versus false-positive rate (FPR), where TPR is on the y-axis and FPR on the x-axis. The AUC metric is used to calculate AUC, and it describes the separability measurement or degree. It informs how the model can identify among classes.

$$Accuracy = \frac{(TN + TP)}{TN + TP + FN + FP} \tag{12}$$

$$Recall = \frac{TP}{FN + TP} \tag{13}$$

$$Precision = \frac{TP}{FP + TP} \tag{14}$$

4. Experiments Results and Discussion

This section presents and discusses the proposed method 1D CNN-BiLSTM, which is validated on two public datasets, namely, Statlog and Cleveland. After that, the classification performance of our method is compared with other machine learning algorithms. Moreover, the implementation of Bayesian optimization on 1D CNN-BiLSTM hyperparameters is discussed.

The proposed architecture is trained for 500 epochs with a batch size of 32. The binary cross-entropy function is used as a loss function to calculate the loss between the true value and the predicted value. This function has to be minimized using some optimization algorithm to achieve convergence. The Adam optimization algorithm is employed for training as it has a faster convergence time and does not zigzag around the local minima.

4.1 Experimental setup

In this section, we evaluate the efficacy of the proposed model by implementing our experiments on two datasets, namely, Statlog and Cleveland datasets. All the computations are conducted on intel i7 CPU and 16 GB RAM. It also has nvidia 2070 GPU, which helps in training the method faster. Moreover, the Python programming software packages scikit-learn and Keras are used for the experiments.

4.2 Results of the Statlog dataset

The model was applied on the Statlog heart-disease dataset, which had 13 features. All the 270 heart disease records of the dataset were considered. In the experiment, we performed the train/test holdout validation. The data were split as 70% for training and 30% for testing. The model was trained on 189 records and tested on the remaining 81 as unseen data. The primary reason behind using this distribution is to satisfactorily compare our approach with those in other researches on the same dataset. We ran the same experimental procedure five times, following which the mean of the five results was calculated. Fig. 4 depicts the training and validation accuracies of the single CNN across different learning epochs. After that, Table 2 compares the results of using 1D CNN with and without BiLSTM for the Statlog dataset. The results show that 1D CNN with BiLSTM is better than the other without BiLSTM. Moreover, Table 3 compares the results of this proposed method with other researches. The proposed approach achieves better classification results than those of most methods. The experimental results on the Statlog dataset confirm that the proposed approach achieves the accuracy rates of 90.91% and 81.48% for the training set and test set, respectively.

Table 2. Compare the proposed method with 1D CNN on the Statlog dataset.

Model	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
1D CNN	90.15	81.18	**83**	82	82
1D CNN-BiLSTM (Our)	**90.91**	**82.72**	82	**83**	**83**

Table 3. Benchmarking our method with others in the literature on the Statlog dataset.

Study	Method	Acc (%)
El-Bialy et al. [30]	C4.5 algorithm and fast decision tree	76.60
Long et al. [31]	Chaos-based firefly algorithm, rough set & type-2 fuzzy logic	78.78
Our method	**1D CNN-BiLSTM**	**82.72**

Fig. 4. Training and validation accuracies of the single CNN across different learning epochs.

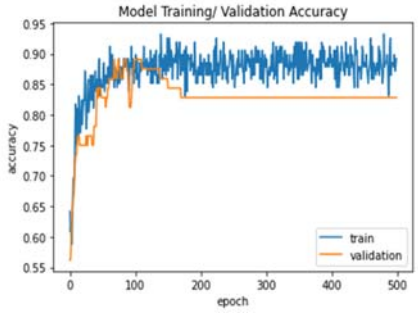

Fig. 5. Training and validation accuracies of the single CNN across different learning epochs.

4.3 Results of the Cleveland dataset

The model was applied to the Cleveland heart disease dataset, which had 13 features. All the 303 heart disease records of the dataset were considered. In the experiment, we performed the train/test holdout validation. The data were split as 70% for training and 30% for testing. The model was trained on 212 records and tested on the remaining 91 as unseen data. The primary reason behind using this distribution is to satisfactorily compare our approach with those in other researches on the same dataset. We ran the same experimental procedure five times, following which the mean of the five results was calculated. The training and validation accuracies of the single CNN across different learning epochs is shown in Fig. 5. Furthermore, Table 4 compares results of using 1D CNN with and without BiLSTM for the Cleveland dataset. The results show that 1D CNN with BiLSTM is better than the other without BiLSTM. In addition, Table 5 compares the results of this proposed method with those of recent researches. The proposed approach achieves better classification results than those of most methods. The experimental results on the Cleveland dataset confirm that the proposed approach achieves the accuracy rates of 90.51% and 89.01% for the training set and test set, respectively.

Table 4. Compare the proposed method with 1D CNN on the Cleveland dataset.

1D CNN Model	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
without Bi-LSTM	89.86	87.90	88	87.5	87.5
With BiLSTM	**90.45**	**89.01**	**89**	**88**	**89**

Table 5. Benchmarking our method with others in the literature on the Cleveland dataset.

Study	Method	Acc (%)
Harkulkar et al. [32]	CNN algorithm	75.2
Chu-Hsing et al. [11]	CNN	80
Paul et al. [33]	Correlation coefficient, GA & fuzzy rules	80
Vivekanandan and Iyengar [34]	Integrated model of fuzzy AHP & ANN	83
Mohammed et al.[10]	Hybrid model by combining ANN, SVM, and NB	88
Gokulnath and Shantharajah [35]	GA & SVM	88.34
Our method	**1D CNN-BiLSTM**	**89.01**

4.4 Comparative Experiments Results

The epoch indicates how often the training set is chosen to update the weights. The model's ability to generalize learning improves as the number of epochs increases. Using a high number of epochs may result in an overfitting issue

on the training set, which will result in the model performing poorly on the validation or test sets. As such, it is critical to determine the optimal number of epochs. Tables 6 and 7 present the empirical results of implementing different epochs for the proposed method on the Cleveland and Statlog datasets, respectively, where the performance of classification increases substantially as the number of epochs increases. After that, the empirical findings in Tables 8 and 9 demonstrate that various activation functions with final layer functions are used for the proposed method on the Cleveland and Statlog datasets, respectively. It has been found that (sigmoid + elu) achieved the highest while Softmax + (elu/relu) achieved the least in test accuracy, precision, recall, and F1 score. Elu does not suffer from the problem of vanishing gradients and exploding gradients. Elu dose not suffer from the problems of dying neurons and has proved to be better compared to relu. Table 10 compares the results of the proposed method for Cleveland dataset with other models. We compare the performance measures on accuracy, precision, recall and f1 score for logistic regression, SVM, Naïve bayes, KNN, Decision trees and RF. From the different classifiers results presented in Table 10, we can see that SVM and KNN achieve higher accuracies, 89.62% and 85.38%, respectively. In contrast, decision trees achieved the least accuracy, 83.60%. Finally, our proposed method 1D CNN-LSTM achieved 90.45% in accuracy. It has been found that our proposed method overcomes the other classifiers and increases the average accuracy by 0.83% compared to the SVM. The experimental results of the proposed method approve the efficiently of this deep learning technique in analyzing data and extracting the important features necessary for prediction.

Table 6. Performance of different number of epochs for the proposed method on the Cleveland dataset.

Epochs	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
50	81.76	81.32	83	83	82
100	87.84	87.91	89	87	88
500	**90.45**	**89.01**	**89**	**88**	**89**

Table 7. Performance of different number of epochs for the proposed method on the Statlog dataset.

Epochs	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
50	83.33	80.25	80	79	80
100	88.64	81.48	82	82	81
500	**90.91**	**82.72**	82	**83**	**83**

Table 8. Performance of using different activation and final layer functions for the proposed method on the Cleveland dataset.

Activation function	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
Softmax + (elu/relu)	54.05	54.95	27	50	35
Sigmoid + relu	**92.57**	86.81	88	86	86
Sigmoid + elu	90.45	**89.01**	**89**	**88**	**89**

Table 9. Performance of using different activation and final layer functions for the proposed method on the Statlog dataset.

Activation function	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
Softmax + (elu/relu)	54.05	54.95	27	50	35
Sigmoid + relu	**92.57**	86.81	88	86	86
Sigmoid + elu	**90.91**	**82.72**	82	**83**	**83**

Table 10. Compare performance for the proposed method with other models on the Cleveland dataset.

Model	Training Acc (%)	Test Acc (%)	Precision (%)	Recall (%)	F1 Score (%)
Logistic regression	85.38	80.22	80	80	80
SVM	89.62	79.12	78	79	77
Naive Bayes	85.38	81.32	81	82	82
KNN	88.68	84.62	84	84	84
Decision trees	83.60	79	80	79	79
RF	84.50	81	81	81	81
Our	**90.45**	**89.01**	**89**	**88**	**89**

5. Conclusion

The goal of this paper is to enhance the prediction of early diagnosis of heart disease. This paper presents a 1D CNN-BiLSTM approach to increase the accuracy of heart-disease diagnosis. The proposed approach achieved high accuracy of 89.01% and 82.72% on the Cleveland and Statlog datasets, respectively. After that, the results of the proposed method with LSTM are compared with the results without using LSTM and found that it outperforms in the accuracy. Moreover, it outperformed the existing state-of-the-art methods on the same datasets. Furthermore, we protected

our model from overfitting by using the dropout layer in CNN architecture. Hence, our experimental results confirmed that the proposed approach enhanced the decision-making process of the practitioners during heart disease diagnosis.

References

[1] S. Roostaee, H. R. J. J. o. E. Ghaffary, and C. E. Innovations, "Diagnosis of heart disease based on meta heuristic algorithms and clustering methods," vol. 4, pp. 105-110, 2016.

[2] D. Banerjee, C. Thompson, C. Kell, R. Shetty, Y. Vetteth, H. Grossman, *et al.*, "An informatics-based approach to reducing heart failure all-cause readmissions: the Stanford heart failure dashboard," *Journal of the American Medical Informatics Association,* vol. 24, pp. 550–555, May. 2017.

[3] E. O. Olaniyi, O. K. Oyedotun, A. Helwan, and K. Adnan, "Neural network diagnosis of heart disease," in *2015 International Conference on Advances in Biomedical Engineering (ICABME)*, 2015, pp. 21-24.

[4] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. Available: http://archive.ics.uci.edu/ml

[5] G. Subbalakshmi, K. Ramesh, M. C. J. I. J. o. C. S. Rao, and Engineering, "Decision support in heart disease prediction system using naive bayes," vol. 2, pp. 170-176, 2011.

[6] N. A. Sundar, P. P. Latha, M. R. J. I. j. o. e. s. Chandra, and a. technology, "Performance analysis of classification data mining techniques over heart disease database," vol. 2, pp. 470-478, 2012.

[7] V. Chaurasia, S. J. I. J. o. A. C. S. Pal, and I. T. Vol, "Data mining approach to detect heart diseases," vol. 2, pp. 56-66, 2014.

[8] K. Revett, F. Gorunescu, A.-B. Salem, and E.-S. El-Dahshan, "Evaluation of the feature space of an erythematosquamous dataset using rough sets," *Annals of the University of Craiova-Mathematics and Computer Science Series,* vol. 36, pp. 123-130, 2009.

[9] S. M. Saqlain, M. Sher, F. A. Shah, I. Khan, M. U. Ashraf, M. Awais, *et al.*, "Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines," *Knowledge and Information Systems,* vol. 58, pp. 139–167, Jan. 2019.

[10] M. J. A. Junaid and R. Kumar, "Data Science And Its Application In Heart Disease Prediction," in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, 2020, pp. 396-400.

[11] C.-H. Lin, P.-K. Yang, Y.-C. Lin, and P.-K. Fu, "On Machine Learning Models for Heart Disease Diagnosis," in *2020 IEEE 2nd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*, 2020, pp. 158-161.

[12] P. Ghosh, S. Azam, M. Jonkman, A. Karim, F. J. M. Shamrat, E. Ignatious, *et al.*, "Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms With Relief and LASSO Feature Selection Techniques," *IEEE Access,* vol. 9, pp. 19304-19326, 2021.

[13] S. Y. El-Bakry, E.-S. El-Dahshan, and M. El-Bakry, "Total cross section prediction of the collisions of positrons and electrons with alkali atoms using Gradient Tree Boosting," *Indian Journal of Physics,* vol. 85, pp. 1405-1415, 2011.

[14] J. J. N. n. Schmidhuber, "Deep learning in neural networks: An overview," vol. 61, pp. 85-117, 2015.

[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, *et al.*, "Backpropagation applied to handwritten zip code recognition," vol. 1, pp. 541-551, 1989.

[16] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, *et al.*, "A deep convolutional neural network model to classify heartbeats," vol. 89, pp. 389-396, 2017.

[17] A. J. N. n. Karpathy, "Cs231n convolutional neural networks for visual recognition," vol. 1, 2016.

[18] I. A. J. T. M. o. E. Dmitrievich, S. o. T. R. F. M. I. o. Physics, and Technology, "Deep Learning in information analysis of electrocardiogram signals for disease diagnostics," 2015.

[19] M. Zubair, J. Kim, and C. Yoon, "An automated ECG beat classification system using convolutional neural networks," in *2016 6th international conference on IT convergence and security (ICITCS)*, 2016, pp. 1-5.

[20] B. Pourbabaee, M. J. Roshtkhari, K. J. I. T. o. S. Khorasani, Man,, and C. Systems, "Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients," vol. 48, pp. 2095-2104, 2018.

[21] F. A. Gers, J. Schmidhuber, and F. J. N. c. Cummins, "Learning to forget: Continual prediction with LSTM," vol. 12, pp. 2451-2471, 2000.

[22] J. Schmidhuber and S. J. N. C. Hochreiter, "Long short-term memory," vol. 9, pp. 1735-1780, 1997.

[23] Ö. J. C. i. b. Yildirim and medicine, "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification," vol. 96, pp. 189-202, 2018.

[24] A. Graves and J. J. N. n. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," vol. 18, pp. 602-610, 2005.

[25] L. De Baets, J. Ruyssinck, T. Peiffer, J. Decruyenaere, F. De Turck, F. Ongenae, *et al.*, "Positive blood culture detection in time series data using a BiLSTM network," 2016.

[26] M. M. K. Mamun and A. Alouani, "FA-1D-CNN Implementation to Improve Diagnosis of Heart Disease Risk Level," in *6th World Congress on Engineering and Computer Systems and Sciences*, 2020, pp. 122.1-122.9.

[27] S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, and J. J. P. R. X. Shiomi, "Designing nanostructures for phonon transport via Bayesian optimization," vol. 7, p. 021024, 2017.

[28] J. J. J. o. G. O. Mockus, "Application of Bayesian approach to numerical methods of global and stochastic optimization," vol. 4, pp. 347-365, 1994.

[29] D. P. Kingma and J. J. a. p. a. Ba, "Adam: A method for stochastic optimization," 2014.

[30] R. El-Bialy, M. A. Salamay, O. H. Karam, and M. E. Khalifa, "Feature analysis of coronary artery heart disease data sets," *Procedia Computer Science,* vol. 65, pp. 459–468, 2015.

[31] N. C. Long, P. Meesad, and H. Unger, "A highly accurate firefly based algorithm for heart disease prediction," *Expert Systems with Applications,* vol. 42, pp. 8221–8231, Nov. 2015.

[32] N. Harkulkar, S. Nadkarni, and B. Patel, "Heart Disease Prediction using CNN, Deep Learning Model."

[33] A. K. Paul, P. C. Shill, M. R. I. Rabin, and M. A. H. Akhand, "Genetic algorithm based fuzzy decision support system for the diagnosis of heart disease," in *5th International Conference on Informatics, Electronics and Vision (ICIEV)*, Dhaka, Bangladesh, 2016, pp. 145–150.

[34] T. Vivekanandan and N. C. S. N. Iyengar, "Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," *Computers in Biology and Medicine,* vol. 90, pp. 125-136, Nov. 2017.
[35] C. B. Gokulnath and S. P. Shantharajah, "An optimized feature selection based on genetic approach and support vector machine for heart disease," *Cluster Computing,* pp. 1–11, Mar. 2018.

Mohamed G. El-Shafiey received his B.SC. degree in information system from the Faculty of Computers and Information, Mansoura University, Egypt, in 2003. He worked in different software Companies in Egypt and KSA as a senior software engineer and team leader from 2004 till now. Currently, he is pursuing M.Sc. in software Engineering at Egyptian E-Learning University, Cairo, Egypt. His current research interests include software engineering, machine learning, and data mining.
Email: mismailaly@eelu.edu.eg

Ahmed Hagag received B.Sc. (Honours) degree in mathematics and computer science from the Faculty of Science, Menoufia University, Egypt, in 2008, and his M.Sc. degree in computer science from the same university, in 2013. He received his Ph.D. degree in computer science from the School of Computer Science and Technology, Harbin Institute of Technology, China, in 2017. He is currently a lecturer in the Faculty of Computers and Artificial Intelligence, Benha University. He has authored several technical journal and conference papers. His current research interests include image processing, deep learning, remote sensing image interpretation, especially compression, classification, and wireless communication.
Corresponding author. Email: ahagag88@gmail.com

El-Sayed A. El-Dahshan received his B.Sc. degree in physics & computer science from AinShams University Cairo, Egypt, in 1986. He received his M.Sc. degree in microwaves from the same university, in 1990. He received his Ph.D. degree in thin films technology, in 1998. Since 2009, he has been the manager of the AinShams Center for the Egyptian e-learning University (EELU). He is currently a professor of scientific computing at EELU and of computational physics at AinShams University. His research interests include wavelet theory and its applications in the fields of signal and image processing, as well as optimisation techniques.
Email: seldahshan@eelu.edu.eg

Manal A. Ismail is a professor with the Faculty of Engineering, Cairo University, Cairo, Egypt. Her research interests include software engineering, artificial intelligence, data mining and knowledge discovery, and distance education and adaptive hypermedia. She graduated from the Department of Electronics & communications Faculty of Engineering of Cairo University, PhD, Ain Shams University, Faculty of Engineering.
Email: manal_shoman@yahoo.com