Research Article

Tamilselvi Kesavan* and Ramesh Kumar Krishnamoorthy

An efficient recurrent neural network with ensemble classifier-based weighted model for disease prediction

https://doi.org/10.1515/jisys-2022-0068
received October 30, 2021; accepted May 07, 2022

Abstract: Day-to-day lives are affected globally by the epidemic coronavirus 2019. With an increasing number of positive cases, India has now become a highly affected country. Chronic diseases affect individuals with no time identification and impose a huge disease burden on society. In this article, an Efficient Recurrent Neural Network with Ensemble Classifier (ERNN-EC) is built using VGG-16 and Alexnet with weighted model to predict disease and its level. The dataset is partitioned randomly into small subsets by utilizing mean-based splitting method. Various models of classifier create a homogeneous ensemble by utilizing an accuracy-based weighted aging classifier ensemble, which is a weighted model's modification. Two state of art methods such as Graph Sequence Recurrent Neural Network and Hybrid Rough-Block-Based Neural Network are used for comparison with respect to some parameters such as accuracy, precision, recall, $f1$-score, and relative absolute error (RAE). As a result, it is found that the proposed ERNN-EC method accomplishes accuracy of 95.2%, precision of 91%, recall of 85%, $F1$-score of 83.4%, and RAE of 41.6%.

Keywords: disease prediction, neural network, ensemble classifier, COVID-19, preprocessing, weighted model

1 Introduction

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is a novel coronavirus that causes coronavirus disease (COVID-19) in the respiratory tract [1]. This epidemic causes serious disasters across the world at an instance and affects nearly 216 nations, with 44 million cases and 1.1 million deaths, according to the statistics of WHO on November 02, 2020 [2]. Because of insufficient amenities for intensive care units, severe illness has reached a point, where health systems, including those in developed countries, have failed. Medical sciences adopted machine learning method for this disease prediction [3]. To be specific, RT-PCR is the gold standard in COVID-19 prediction, but it consumes a lot of time, laboratory, and complex manual process, which are in shortage. Moreover, RT-PCR sensitivity varies highly and the results are not clear and consistent, and in China its initial findings are with low relative sensitivity [4]. Its succeeding results produce increased variable positive rate based on the specimen gathered, and the positive rate also decreases with time after the initial symptom. The process of optimization and improvement of these results are proceeded by the researchers. Machine learning tasks are enhanced by an approach called ensemble learning [5]. Individual classifier set is combined together in an ensemble classifier using a process similar

*** Corresponding author: Tamilselvi Kesavan,** Research and Development, Bharathiar University, Coimbatore, Tamil Nadu, India, e-mail: tamilkselvi21@gmail.com
Ramesh Kumar Krishnamoorthy: Department of Computer Science, Bharathiar University, Coimbatore, Tamil Nadu, India, e-mail: rameshkumark.dr@gmail.com

to majority voting, which is the combination of component prediction. This ensemble classifier performs better compared to the traditional classifiers [6]. A single base learner or algorithm is present in the members of homogeneous ensemble learning [7]. In the meantime, different structures are associated with members. However, various base learners are present in the members of heterogeneous ensemble learning [8]. Many machine learning approaches are motivated towards the enhancement of COVID disease risk prediction and performance of classification. Recurrent Neural Network with ensemble classifiers are mainly used for various disease predictions such as heart disease, Alzheimer's disease, etc. VGG means Visual Geometry Group with multiple layers in convolution neural network architecture. The VGG-16 or VGG-19 consists of 16 and 19 convolution layers. Alexnet is the ImageNet Classification with Deep Convolutional Neural Networks, which is used in the machine learning field for classification. Based on this, this article proposes a type of homogeneous ensemble learning approach. The contribution of this work is as follows:

- To construct an Efficient Recurrent Neural Network with Ensemble Classifier (ERNN-EC) by utilizing VGG-16 and Alexnet with a weighted model for predicting the disease and its level.

This article is presented as follows: Section 1 defines the background of COVID disease prediction and the role of neural network-based classification in disease prediction. In Section 2, the literature for various disease predictions using neural network classification is given. Section 3 explains the proposed neural network for disease prediction. In Section 4, the experimental analysis is given with graphs by comparing them with two standard methods. Finally, Section 5 concludes the article by presenting the conclusion and future work.

2 Related works

In, ref. [9], an artificial neural network-dependent classifier utilizing multilayer perceptron with backpropagation algorithm was developed for peak event prediction (peak demand days) in patients with respiratory diseases such as asthma and chronic obstructive pulmonary disease. Peak event class achieves a precision of 77.1% and a recall of 78.0%, whereas nonpeak events achieves a precision of 83.9% and a recall of 83.2%. The entire system accuracy is 81.0%. Graph Sequence Recurrent Neural Network (GS-RNN) is proposed in ref. [10] in which FoG patterns are characterized through the emerging graph recurrent cells, which carry dynamic structure's graph sequences as inputs. Intended for unavailable prior edge annotation cases, a data-driven dependent adjacency estimation approach is also developed and this first approach of vision-based FoG detection utilizing DNN that is intended for dynamic structure's sequences of graph. An appropriate classifier is selected in a block-based neural network (BBNN) in ref. [11] because of its capability of evolving internal structures and dynamic environment adaptability. Application characteristics are incorporated systematically in this architecture for structuring Hybrid rough-Block-Based Neural Network (R-BBNN). Hybrid particle swarm optimization having wavelet mutation is the global training algorithm developed for optimizing the parameter of R-BBNN. Improved CNN-based trained network named VggNet16-wbn is presented in ref. [12], which is the derivation of VggNet16 (pre-trained CNN algorithm) (Table 1).

Table 1: Comparison of GS-RNN and the hybrid R-BBNN

S. No	GS-RNN	Hybrid R-BBNN
1	GS-RNN optimization is based on weight regularization, activation function optimization, and batch normalization	R-BBNN optimization is a weight-based genetic algorithm
2	GS-RNN estimates the sample value at time frame utilizing the values of multiple frames	R-BBNN model includes some restrictions such as 2D array and integer weights for estimating sample values
3	The gradient problem vanishing is mitigated by this algorithm	The engineering problems such as pattern classification and mobile robot control

Transfer learning, which is a deep learning approach, as well as automated ocular detection method, is discussed in this article. A CNN-based trained network architecture was present depending on network assessment from six CNN pre-trained networks for detecting conjunctiva. Dual-input neural network presented in ref. [13] was used in the feature extraction and deep learning method integration. Multiple domains are used for the extraction of ECG and PCG features and ratio of information gain is used for significant feature selection. In this article, signals of ECG and decayed PCG are integrated as five-channel signal. Automated detection of kidney disease using sensing method is presented in ref. [14]. The concentration of salivary urea is observed the disease detection. The saliva sample's urea levels are monitored by this approach. Moreover, the raw signals are analyzed from the sensor and the 1D DL-CNN algorithm is implemented by incorporating support vector machine (SVM) classifier. Some of the previous methods of leveraged artificial neural networks architectures and several well-performed methods that are used in the statistical approaches, such as LRM, for developing prediction models to estimate the liver failure likelihood are discussed. These statistics approaches permit the researchers to develop the prediction models for predicting results based on the independent variable set. Sometimes, incorrect independent variables are incorporated by the researchers. Likewise, statistical models that appeared susceptible to bullishness and had high prediction power resulted in sampling bias. All these limitations can be easily overcome by ERNN-EC as described in Section 3.

3 System model

The neural network-based disease prediction architecture is constructed in a homogeneous manner as shown in Figure 1. Initially, the datasets are preprocessed and trained using Recurrent Neural Network with Perception Algorithm (RNNPA). Second, the ensemble classifiers, such as VGG-16 and Alexnet, are

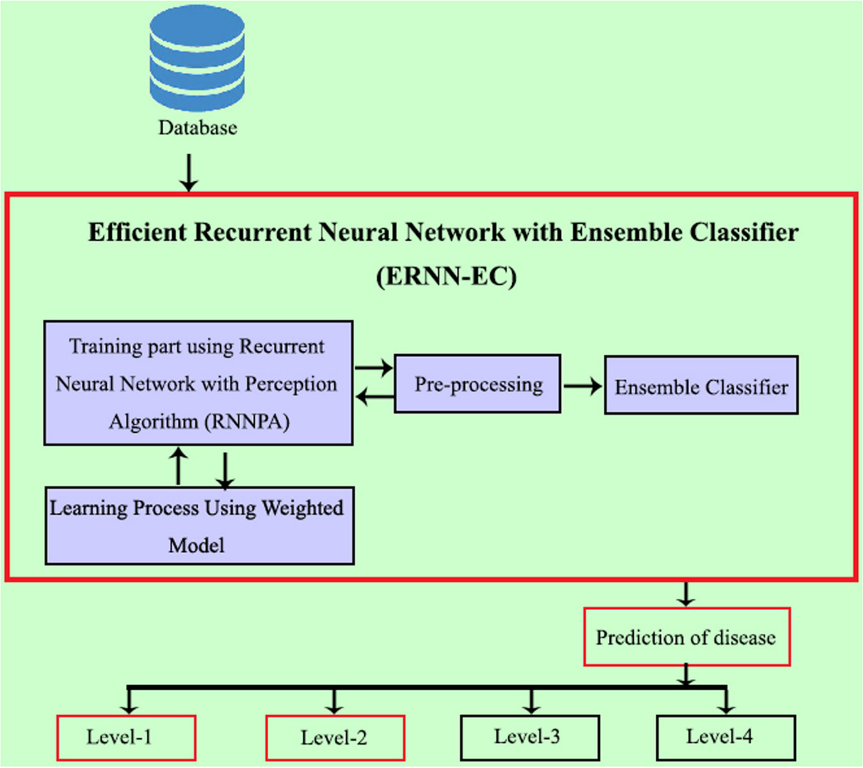
Database
Efficient Recurrent Neural Network with Ensemble Classifier (ERNN-EC)
Training part using Recurrent Neural Network with Perception Algorithm (RNNPA)
Pre-processing
Ensemble Classifier
Learning Process Using Weighted Model
Prediction of disease
Level-1
Level-2
Level-3
Level-4

Figure 1: System architecture for neural network-based disease prediction.

used for better classification with the reduction of error in the output side. Finally, the learning process is done using weighted model for finalizing the level of disease.

4 Dataset description

The clinical data and demographics of the COVID-19-positive patient's clinical data and its demography are included in the dataset. This dataset contains COVID-19 records of 287 patients with binary class labels known as "survived" and "deceased," which are 243 and 44, respectively. The BodyTemp 1 field designates the first temperature of body that is reserved when the patient is admitted to the hospital. The reading of the last temperature of the body reserved prior to the discharge of patient is indicated in BodyTemp 2. Likewise, breath shortness is represented by shortness of breath, chronic disease diabetes mellitus is represented by chr-dm, hypertension is represented by chr-Htn, cardiovascular diseases are represented by chr-cardiac, and dyslipidemia and cough are represented by chr-dlp.

5 Preprocessing and training using RNNPA

In RNNPA, the neuron's preliminary mission within the input layer is to partition the input signal xi among the neurons in the hidden layer, which is represented in Figure 2. Each *j*th neuron in the hidden layer enhances its *xi* input signals by weighting them with the respective connection strengths *wji* from the input layer, and its output *yj* is determined as a sum function *f*, as provided in the following equation:

$$y(j) = f\left[\sum_{i=0}^{n} w(ji)x(i)\right]. \tag{1}$$

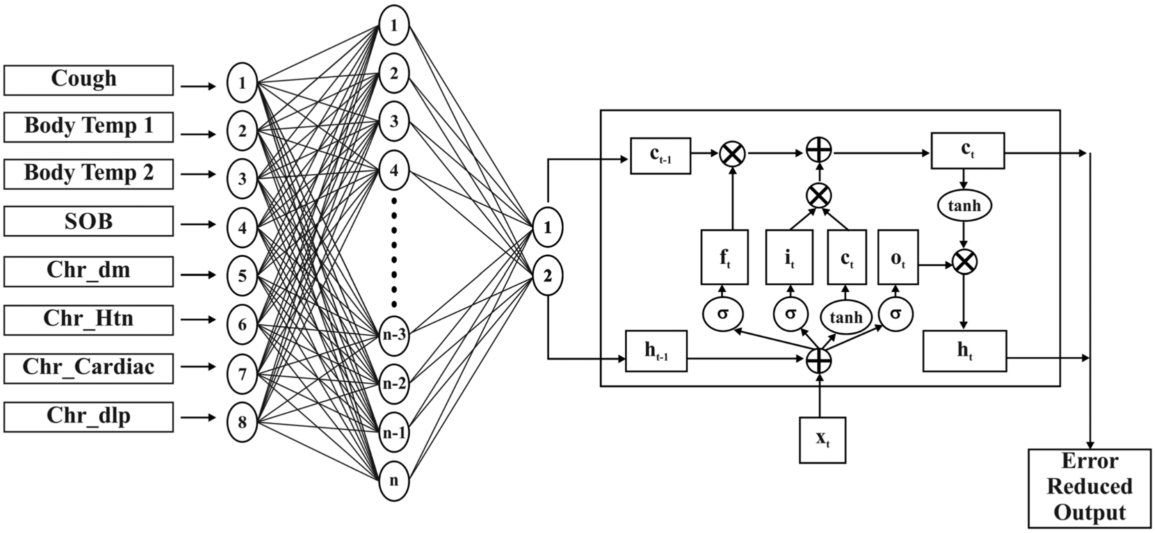

Figure 2: Architecture of recurrent neural network with perception.

The working of RNNPA is summarized as follows:

Step 1: For processing, input layer is provided with input data, from which predicted output is obtained; $X = x1, x2, x3, \ldots, xn - 1, xn$ is the sequence provided. As t is the timestep, the input is $xt \in Rk$. At the same time [15], the hidden state, $ht \in Rd$, is given as shown in the following equation:

$$ht = f(ht - 1, xt) = f(Uht - 1 + Wxt + b). \tag{2}$$

Step 2: The value of error is calculated by the difference between predicted output and actual output.

Step 3: Back-propagation with the Stochastic-based Nelder–Mead method is used in this network to reduce complexity while using the standard back-propagation method. Here, in every iteration, the training instance subset or batch is used. As a result, the error computed is biased as per optimum, but it was performed quickly. Every iteration in the whole training set needs numerous iterations in the small batches, and adjusting weights are provided in equations (3)–(8):

$$it = \beta(Utht - 1 + Wixt + bi), \tag{3}$$

$$ft = \beta(Ufht - 1 + Wfxt + bf), \tag{4}$$

$$0t = \beta(Uoht - 1 + Woxt + bo), \tag{5}$$

$$ct = \tanh(Ucht - 1 + Wcxt + bc), \tag{6}$$

$$ct = ft \cdot ct - 1 + it \cdot ct, \tag{7}$$

$$ht = Ot \cdot \tanh(ct). \tag{8}$$

Step 4: The weights of the output layer node and last hidden layers node weight are weighed for adjusting weights by the network that works backward. Input x with non-linear function sigmoid (σ) or tanh described in equation (9) [16] is triggered in hidden neurons, and linear function $f(x) = x$ is triggered in output neuron.

$$\sigma(x) = \frac{1}{1 + e - x}. \tag{9}$$

Step 5: As back propagation is completed, the forwarding process begins again.

The reset gate with reduced error permits the hidden layer to drop any information without future prediction, and the amount of information that are controlled by the updated gate from the prior hidden layer will remained in the present hidden layer. Particularly, the input vector xt is appended to train this model, and an additional dimension denotes the time as log of the number of days between t, time of event, and, T is date of index. Logarithmic transformation is applied for minimizing the duration's skewed distribution

Input: Training data

Output: Error reduced data

Training data (T) = Weighted Difference (Vector.T1,Vector.T2)

$$(yj, xj)j = \sum_{0}^{n} j < n, \text{ where } xj \text{ and } yj' = 0, 1$$

standardize the weights ωr.

For each and every feature, weak classifier hi is trained.

A classifier hi with error εi is considered to match the weight ωr

$$\varepsilon i = \sum \omega r, j|hi(yj) - xj|nj = 1.$$

```
Set $iter = 0$
While $iter = 0$
For each view
$Hj(x) \rightarrow class\ cj$
$ifHj(x) = wi * cji(x)$
$H_i(x) = H_j(x)$
End if
```

```
Base = Base + Weighted
Trial = Aggregate(Target,Base)
if $MF$(Trial) < $MF$(Base) then
Trial = Base
else
Target = Base
end if
```

6 Ensemble classifier

Two RNN models, VGG-16 and Alexnet, used in this experiment, use 13 convolutional layers and 3 fully connected layers. Input for the neural network is an image with 224 × 224 × 3 size. The filters are 3 × 3 matrices and the stride of which is fixed to 1. The padding size is always 1, while max-pooling is carried out in a pixel window of size 2 × 2, with stride of 2. Rectified Linear Unit (ReLU) layer, which always follows the convolution layer in VGG-16, boosts the nonlinearity of VGG-16 . The convolution layers present within both the pooling layers possess same channel number, kernel size, and stride. Actually, collecting two 3 × 3 convolution layers and three 3 × 3 convolution kernels is equal to a single 5 × 5 and 7 × 7 convolution layer, respectively. Stacking 2 or 3 small convolution kernels works much quicker than a single huge convolution kernel. Moreover, parameter numbers have been minimized. ReLU layers that are inserted between undersized convolution layers are really useful. To eliminate the trade-off problem, transfer learning as well as multiple pre-trained Alexnet is applied. Suitable approaches are required in this Alex-Net model for faster training, and overfitting is prevented because of its complicated structure, several training parameters, and huge training data. As Alex-Net model is built and nonlinearity of ReLU in the data structures directly utilized for making initialized approach is highly consistent with theory and network training began directly from initial point for enhancing the speed of training, local response normalization was carried out in the process known as "near-suppression" operation, which efficiently enhances the performance generalization of this approach by normalizing the local input regions. The proposed framework for ensemble learning incorporates multiple fusion levels of miscellaneous classifiers that are trained on various sets of features. Likewise, the extraction of feature set done by particular approach is processed further to obtain various subsets of features using various methods of feature selection. In the third layer, multiple classifiers' training, base classifiers are trained by using m learning algorithms on every feature set Fi, thus Ei is the preliminary ensemble formed on every n feature sets that are present in primary fusion layer. Finally, primary ensembles n designed on feature set n are bonded further for creating the final ensemble, because the final classification is finished as the final ensemble output are presented in the final fusion layer. The cross-validated loss for base learner j, which is defined first [17], is given as shown in the following equation:

$$R_{cv} = \sum_{v=1}^{v} \sum_{i \in \mathrm{val}(v)}^{n} l(yi, pij), \tag{10}$$

where, in the vth fold, the indices set of observations is indicated by val(v), and the ith observation with prediction is defined as pij, using the base learner j that is trained on the entire data except the fold v.

The task of binary classification is considered simple as it can be easily generalized for multi-class classification and regression. The single algorithms of Super Learner's simple version are first studied using negative (Bernoulli) log-likelihood as loss function and are represented in the following equations:

$$l(y, p) = -[y \log(p) + (1 - y) \log(1 - p)], \tag{11}$$

$$R_{cv} = \sum_{v=1}^{v} \sum_{i \in \mathrm{val}(v)}^{n} yi \log\left(\sum_{j=1}^{m} a_j p_{ji} \right) + (1 - yi) \log(i - a_j p_{ji}), \tag{12}$$

where, ith unit predicted probability from jth base learner is denoted by pji, which is trained on the entire data except vth fold. For K-class classification having softmax output, such as neural networks, the score level in ensemble is given as shown in the following equation:

$$Pzi(a) = \frac{\exp\left(\sum_{j=1}^{m} aj.\ sj[j, z]\right)}{\sum_{k=1}^{k} \exp\left(\sum_{j=1}^{m} aj.\ sj[j, z]\right)}. \tag{13}$$

where the ith unit ensemble prediction and class z with weight vector "a" is $Pzi(a)$. mXK matrix is si and model j and class k score is stand by $si[j, k]$.

7 Learning process using weighted model

The variable $x(i)$'s weighted mean is given as shown in the following equation:

$$X(i) = \frac{1}{N}\sum_{i=1}^{N} wixij, \quad wi' > 0. \tag{14}$$

The weighted mean occupies more contribution of data elements having increased weight, and it occupies less contribution of data elements with low weights. There should not be any negative weights, and there are zero weights. The algorithm of data partitioning is developed by introducing the randomization, i.e., at every partitioning, a random variable dataset is used by this algorithm.

In the latter, individual outputs and different weights are combined to take any of the forms described below [18]:

- For every classifier, particular weight is assigned, and the combined output for class cji is given as shown in the following equation:

$$Hj(x) = \frac{1}{T}\sum_{i=1}^{T} wi * cji(x), \tag{15}$$

where **Hj** is the classifier with the assigned weight **wi**.

- For every classifier hi, class-specific weight $w(i)j$ is assigned, and the combined output for class **cji** is given as shown in the following equation:

$$Hj(x) = \frac{1}{T}\sum_{i=1}^{T} wi(j) * cji(x). \tag{16}$$

- For every sample in a class of every classifier, weight is applied, and the combined output of **cji** is given as shown in the following equation:

$$Hj(x) = \sum_{i=1}^{T} n \sum_{i=1}^{T} wi(j) * Hj(x), \tag{17}$$

where $wi(j)$ is the weight assigned to instance xk of cj for Hj.

8 Detection of disease and its level

Later, in the training of classifiers, an equivalent process is carried out for the testing set: the feature selected is acquired from the training set. The mean values are obtained by imputing the missing values from the training set and processing the z-score depending on the consistent mean and standard deviation features from the training set.

At last, the trained classifier's output on the testing set with the real labels is compared for the performance evaluation. Every result of validation and associated attribute subsets are registered in a list that is sorted as per metrics. Ties are solved in the following order: Level 1, very low; Level 2, low; Level 3, Moderate; and Level 4, high. In the conclusion of process, top subset is the best subset. During the testing stage, the algorithm is applied on both the training union and the dataset of testing, acquiring the attribute subset's rank. In the final case, the top of the rank subset is called the optimal attribute subset (OAS). It is evident that the final rank will not be obtainable in practice, as the dataset for testing denotes samples which are not found in samples obtained in the future. The attribute rank combinations creat the OAS that is utilized for defining a performance metric, which is utilized only for the purpose of evaluation.

9 Performance analysis

Performance evaluation was carried out using the parameters such as accuracy, Precision, recall, $F1$-score, and relative absolute error (RAE). These parameters are contrasted with two existing methods, such as GS-RNN and Hybrid R-BBNN, together with the proposed ERNN-EC (Table 2).

Table 2: Parameters for operation

Operational parameters	Values
Size of the batch	15
Rate drop for learning	1×10^{-2}
Period of learn rate drop	30
Hidden layers	8
Epochs	160
Batch normalization epsilon	10^{-4}
Initial learn rate	1×10^{-2}

The investigation parameters, including accuracy, precision, recall, $F1$-score and RAE, are discussed as follows:

- **Accuracy**: The network model capability is generally predicted by accuracy. The classifier capacity is computed by true positive (TP) and true negative (TN), which are used for the calculation of lack and being there of disease. The false prediction amount obtained by the model is recognized by false negative (FN) and false positive (FP). The accuracy formula is provided in the following equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{18}$$

Table 3 represents the accuracy of comparison among the existing GS-RNN and R-BBNN methods with the ERNN-EC method.

Table 3: Analysis of accuracy

Number of samples	GS-RNN [10]	R-BBNN [11]	ERNN-EC [proposed]
200	88	90	92
400	90	91	94
600	92	93	95
800	93	95	97
1,000	94	96	98

Figure 3 illustrates the comparison of accuracy between existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method, where X axis represents the number of datasets utilized for examination and accuracy acquired in percentage is represented in Y axis. When compared, existing GS-RNN and R-BBNN methods achieve 91.4 and 93%, whereas the proposed ERNN-EC method achieves 95.2% of accuracy which is 3.8% better than GS-RNN and 2.2% better than R-BBNN.

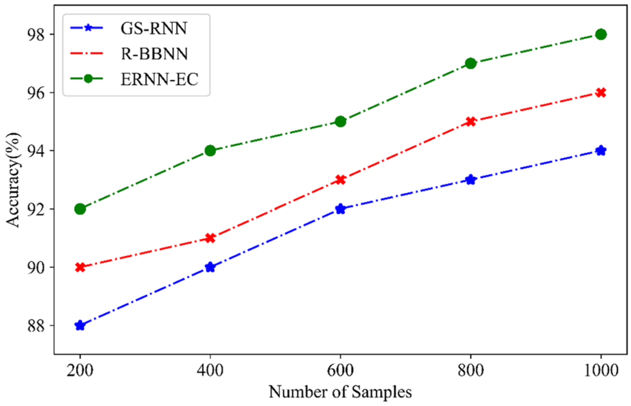

Figure 3: Comparison of accuracy.

Precision: The model of disease classification's entire achievement is indicated by Precision. The classification function's likelihood is forecasted for getting a positive rate when the disease is present. The true positive (TP) amount is recognized and it is calculated as given in the following equation:

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{19}$$

Table 4 shows the comparison of precision between the existing GS-RNN and R-BBNN with the proposed ERNN-EC method.

Table 4: Analysis of precision

Number of samples	GS-RNN [10]	R-BBNN [11]	ERNN-EC [proposed]
200	82	84	87
400	84	86	88
600	85	89	92
800	87	91	93
1,000	90	92	95

Figure 4 illustrates the comparison of precision between the existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method, where X axis represents the number of datasets utilized for evaluation and precision acquired in percentage is represented in Y axis. When compared, the existing GS-RNN and R-BBNN methods achieve 85.6 and 88.4%, whereas the proposed ERNN-EC method achieves 91% of precision which is 5.4% better than GS-RNN and 2.6% better than R-BBNN.

Recall designates the classifier likelihood that attains negative result when there is no disease and also called true negative (TN) rate, which is computed by the following equation:

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{20}$$

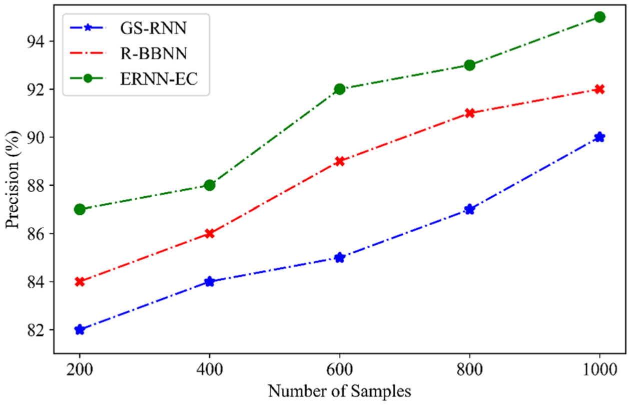

Figure 4: Comparison of precision.

Table 5 represents the comparison results of recall among the existing GS-RNN and R-BBNN with the proposed ERNN-EC method.

Table 5: Analysis of recall

Number of samples	GS-RNN [10]	R-BBNN [11]	ERNN-EC [proposed]
200	74	77	80
400	76	79	82
600	79	82	85
800	80	85	88
1,000	83	87	90

Figure 5 illustrates the comparison of recall between the existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method, where X axis represents the number of datasets used for analysis and the recall acquired in percentage is represented in Y axis. When compared, the existing GS-RNN and R-BBNN methods achieve 78.4 and 82%, whereas the proposed ERNN-EC method achieves 85% of recall which is 6.6% better than GS-RNN and 3% better than R-BBNN.

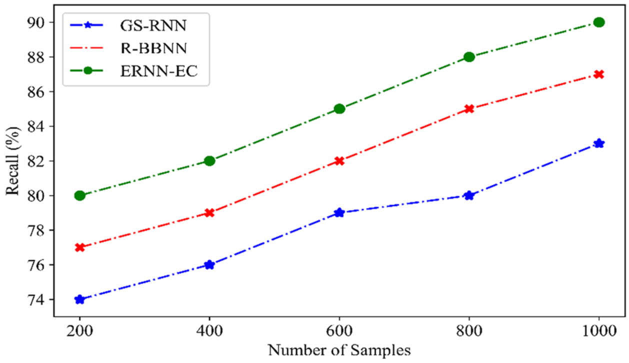

Figure 5: Comparison of recall.

***F*1-score** is a feat for establishing prediction performance. It is built by evaluating the precision and recall's harmonic portion. If a score value is computed as 1, then it is restrained as highly brilliant, and if results are bad, then the score value is 0. The true negative rate is not considered by $F1$-score in its account. The $F1$-score is computed as shown in the following equation:

$$F1\text{-score} = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}}. \tag{21}$$

Table 6 shows the comparison of $F1$-score between the existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method.

Table 6: Analysis of $F1$-score

Number of samples	GS-RNN [10]	R-BBNN [11]	ERNN-EC [proposed]
200	71	75	79
400	73	79	81
600	75	81	84
800	78	82	85
1,000	81	85	88

Figure 6 represents the $F1$-score's comparison between the existing GS-RNN and R-BBNN methods and proposed ERNN-EC method, where X axis represents the number of datasets utilized for evaluation and Y axis indicates the $F1$-score acquired in percentage. When compared, the existing GS-RNN and R-BBNN methods achieve 75.6 and 80.4%, whereas the proposed ERNN-EC method achieves 83.4% of $F1$-score, which is 7.8% better than GS-RNN and 3% better than R-BBNN.

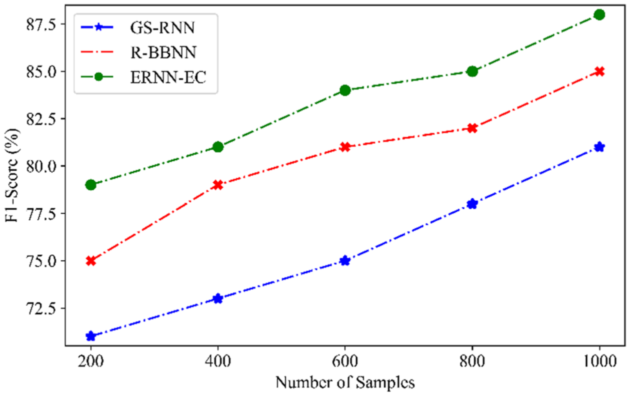

Figure 6: Comparison of $F1$-score.

- **RAE**

The comparison of a mean error (residual) to errors obtained by an unimportant or naive model ratio is known as RAE. The formula of RAE is given as shown in the following equation:

$$\mathrm{RAE} = \frac{\sum_{i=1}^{n}(pi - Ai)^2}{\sum_{i=1}^{n}Ai}. \tag{22}$$

Table 7 shows the comparison of RAE between the existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method.

Table 7: Analysis of RAE

Number of samples	GS-RNN [10]	R-BBNN [11]	ERNN-EC [proposed]
200	46	40	33
400	48	42	37
600	51	45	42
800	53	49	46
1,000	55	52	50

Figure 7 illustrates the comparison of RAE between the existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method, where X axis indicates the number of datasets utilized for evaluation and RAE acquired in percentage is represented as Y axis. When compared, the existing GS-RNN and R-BBNN methods achieve 50.6 and 45.6%, whereas the proposed ERNN-EC method achieves 41.6% of RAE, which is 9% better than GS-RNN and 4% better than R-BBNN.

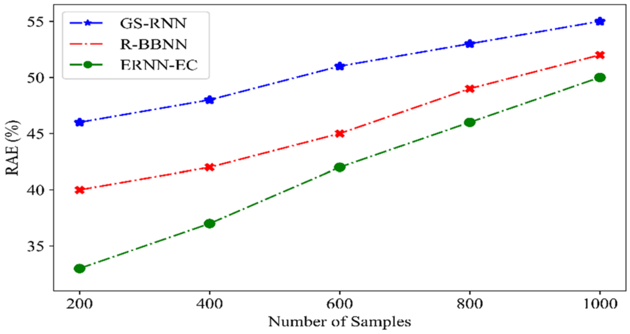

Figure 7: Comparison of RAE.

Table 8 shows the overall comparison between the existing GS-RNN and R-BBNN methods and the proposed ERNN-EC method.

Table 8: Overall comparative analysis between existing and proposed methods

Parameters	GS-RNN [10]	R-BBNN [11]	ERNN-EC [proposed]
Accuracy (%)	91.4	93	95.2
Precision (%)	85.6	88.4	91
Recall (%)	78.4	82	85
$F1$-score (%)	75.6	80.4	83.4
RAE (%)	50.6	45.6	41.6

10 Conclusion

In the neural network field, significant role is performed by deep learning and data mining. The core objective is to enhance the accuracy of predictive model. The COVID-19 disease is efficiently predicted in this research. This approach uses ERNN-EC to randomly partition a dataset into smaller subsets based on the classification entered into the model with every partition. An accuracy-based weighted aging classifier is used in the creation of the homogeneous ensemble. This approach is contrasted with the two existing methods, like GS-RNN and Hybrid R-BBNN, in terms of parameters such as accuracy, precision, recall,

$F1$-score, and RAE. As a result, it is found that the proposed ERNN-EC method accomplishes an accuracy of 95.2%, precision of 91%, recall of 85%, $F1$-score of 83.4%, and RAE of 41.6%. The future work concentrates on constructing optimization-based autoregressive models for improving the accuracy of neural networks.

Acknowledgments: All the contents and images used in the articles are cited and acknowledged in the reference section.

Conflict of interest: The authors state no conflict of interest.

References

[1] Li J, Xi L, Wang X, Fang F, Lv X, Zhang D, et al. Radiology indispensable for tracking COVID-19. Diagnostic Intervent Imaging 2020;102(2):69–75.

[2] Salehi S, Abedi A, Balakrishnan S, Gholamrezanezhad A. Coronavirus disease 2019 (COVID-19): a systematic review of imaging findings in 919 patients. Am J Roentgenol. 2020;215(1):87–93.

[3] Wang S, Zha Y, Li W, Wu Q, Li X, Niu M, et al. A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. Europ Respirat J. 2020;56(2):2000775.

[4] Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, et al. Clinical features of patients infected with 2019 Novel Coronavirus in Wuhan China. Te Lancet. 2020;395:497–506.

[5] Guan WJ, Hu Y, Ni ZY. Clinical characteristics of Coronavirus disease 2019 in China. N Engl J Med. 2020;382(18):1708–20.

[6] Shang F, Zhou K, Liu H, Cheng J, Tsang IW, Zhang L, et al. VR-SGD: a simple stochastic variance reduction method for machine learning. IEEE Trans Knowledge Data Eng. Jan. 2020;32(1):188–202. doi: 10.1109/TKDE.2018.2878765.

[7] Ebenuwa SH, Sharif MS, Alazab M, Al-Nemrat A. Variance ranking attributes selection techniques for binary classification problem in imbalance data. IEEE Access. 2019;7:24649–66. doi: 10.1109/ACCESS.2019.2899578.

[8] Rahim R, Murugan S, Priya S, Magesh S, Manikandan R. Taylor based grey wolf optimization algorithm (TGWOA) for energy aware secure routing protocol, Int J Comput Netw Appl (IJCNA). 2020;7(4):93–102.

[9] Khatri KL, Tamil LS. Early detection of peak demand days of chronic respiratory diseases emergency department visits using artificial neural networks. IEEE J Biomed Health Informatics. 2017;22(1):285–90.

[10] Hu K, Wang Z, Wang W, Martens KAE, Wang L, Tan T. Graph sequence recurrent neural network for vision-based freezing of gait detection. IEEE Trans Image Process. 2019;29:1890–901.

[11] San PP, Ling SH, Nguyen H. Evolvable rough-block-based neural network and its biomedical application to hypoglycemia detection system. IEEE Trans Cybernetics. 2013;44(8):1338–49.

[12] Zamani NSM, WMDW Zaki, Huddin AB, Hussain A, Mutalib HA Ali A. Automated pterygium detection using deep neural network. IEEE Access 2020;8:191659–72.

[13] Li H, Wang X, Liu C, Wang Y, Li P, Tang H. Dual-input neural network integrating feature extraction and deep learning for coronary artery disease detection using electrocardiogram and phonocardiogram. IEEE Access 2019;7:146457–69.

[14] Bhaskar N, Manikandan S. A deep-learning-based system for automated sensing of chronic kidney disease. IEEE Sensors Letters 2019;3(10):1–4.

[15] Rajanayagam J, Frank R, Shepherd RW, Lewindon PJ. Artificial neural network is highly predictive of outcome in pediatric acute liver failure. Pediatric Transplant. 2013;17(6):535–42.

[16] Valero D, Bung DB. Artificial Neural Networks and pattern recognition for air-water flow velocity estimation using a single-tip optical fibre probe. J Hydro Environ Res. 2018;19:150–9.

[17] Lai CC, Su KL. Development of an intelligent mobile robot localization system using Kinect RGB-D mapping and neural network. Comput. Electr. Eng. 2018;67:620–8.

[18] Erkaymaz O, Ozer M, Perc M. Performance of small-world feedforward neural networks for the diagnosis of diabetes. Appl Math Comput. 2017;311:22–8.