International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Transfer Learning Approach for Bipolar Disease Detection

[1]R. Saranya, [2]Dr.S. Niraimathi,

Submitted: 06/02/2024 **Revised**: 14/03/2024 **Accepted**: 20/03/2024

Abstract: Bipolar illness is a complicated mental health issue that affects a large section of the world. Effective bipolar illness therapy requires early and precise diagnosis. This work proposes a unique transfer learning strategy for bipolar disorder identification using many modules to improve prediction accuracy. The first module trains a CNN, BiLSTM, and RBF. Deep learning architectures help this module extract relevant characteristics from raw input data. The second module uses a DNN for feature selection to enhance feature representation. The DNN model eliminates superfluous or duplicated data by identifying the most important bipolar disorder diagnosis characteristics. In the third module, transfer learning uses a pre-trained model. Pre-trained models improve bipolar illness prediction by using learnt representations. Transfer learning is modified to include domain knowledge from related activities or datasets. The fourth module implements a RESNET Classification module. RESNET excels in picture categorization; therefore we use it to forecast bipolar disorder by capturing complicated data patterns and correlations. In the fifth module, SGD optimizes the model. By repeatedly adjusting parameters based on a portion of training data, SGD speeds convergence and improves accuracy. Finally, we optimize Levy Flight-based Fruit fly optimization to fine-tune model parameters. This technique optimizes hyper parameters including learning rate, batch size, and regularization for optimal bipolar illness identification.

Keywords: *Bipolar Disorder, BiLSTM, Deep Learning, Resnet, Optimization*

1. Introduction

Severe and persistent mood fluctuations, including manic and depressive episodes, are hallmarks of bipolar disorder [1]. It impacts a substantial section of humanity and significantly impacts people's and families' quality of life [2]. As a major factor in the worldwide burden of illness and having far-reaching effects on people's social and economic well-being, mental disorders pose serious challenges to public health [3]. By 2020, mental diseases will have surpassed all other causes of disability globally, according to the World Health Organization. The World Health Organization estimates that 27% of the population may have mental health issues in their lifetimes, with over 264 million men and women of all ages living with a diagnosable mental condition [4-6]. Indirect expenses account for the vast majority of the more than 210 billion dollars spent on and drug use problems in the US alone [7]. This means that illnesses affecting the central nervous system are more expensive than diabetes and cancer put together and is responsible for 35% of all cases of sickness [8]. An important part in the psychiatric revolution may be played by artificial intelligence (AI). Improvements in early accurate diagnosis and prognosis, the development of novel treatments, and the creation of assistive technologies for longitudinal patient follow-up can all be achieved through the use of multimodal Artificial Intelligence-based approaches and technologies [9-10].

The morbidity and death rate associated with bipolar disorder (BD) is significant. It is a chronic and recurring condition. An estimated 2% of the overall population suffers from this [11]. Mood swings, including despair, manic episodes, and mixed moods, occur throughout the course of illness. There is a substantial socioeconomic cost, a heavy burden and a significant influence on psychosocial functioning and quality of life due to the recurring and chronic character of bipolar illness [12-14]. Mood stabilizing medication therapy greatly reduces the probability of recurrence. The severity of symptoms and the pace of progression to full-blown disease may be mitigated by early pharmaceutical intervention. Nevertheless, individuals often exhibit a lack of awareness about their symptoms and the need of therapy, particularly when it comes to manic episodes [15]. Regular clinical visits are also a part of the present paradigm for treating bipolar illness, although the frequency of control visits is often inadequate. This means that treatment adjustments are typically postponed, and in some cases made throughout a whole episode. Therefore, better methods of avoiding emotional breakdowns are required. A potential solution may include constantly tracking several indicators of sickness progression, such mood, sleep, or exercise, using a smartphone [16].

Our optimised Levy Flight-based Fruit fly optimisation approach is finally provided in the sixth module [17].

[1]*PhD, Research Scholar, Department of Computer Science, NGM College, Pollachi.*
r.saranyacms@gmail.com
[2]*Associate Professor, Department of Computer Science, NGM College, Pollachi.*

Optimising the model's key parameters—including learning rate, batch size, and regularization—for optimal performance in bipolar illness diagnosis may be achieved with the use of this technique [18]. By combining different modules and methodologies, our transfer learning model aims to make bipolar illness detection more accurate and efficient [19]. Through extensive testing and evaluation on a large dataset, we demonstrate the effectiveness of our proposed method in comparison to existing methods [20-22]. Healthcare providers may benefit from this study's results in a number of ways, including improved clinical decision-making, earlier diagnosis, and better treatment and management outcomes for patients with bipolar disorder [23–24].

2. Background Study

Kessing, L. V.et al. [9] All categories of somatic disorders except malignancy were shown to be related with higher rates in this comprehensive population-based nationwide analysis involving registry data from the whole Danish population. On the other hand, viral and parasitic infections, disorders affecting the nervous, digestive, and genitourinary systems, and other illnesses were more common among the asymptomatic siblings of people with bipolar disorder. To combat this, future efforts to prevent the disorder should prioritize environmental factors, such as studying the side effects of psychotropic drugs and unhealthy lifestyle choices. Physical diseases should also be better monitored, detected early, and treated.

Passos, I. C. et al. [13] Risk prediction, customized therapy, and prognosis were all areas that might benefit from more sophisticated computational techniques, and the high death and morbidity rates associated with BD encourage this kind of research. The author of this paper delves into the possibilities of individualized prediction models offered by the industry as a result of big data analytics and machine learning. Although several of the included studies did make use of machine learning techniques, they refrained from using very large datasets. Unrepeated, small-scale investigations have yielded some of the most intriguing findings. Since the area of big data and machine learning in BD was just starting off, it was vital to replicate the findings. New developments in big data analytics and machine learning, however, allow the writer to delve deeply into the "real patient" and all of their complexities.

Ren, Z. et al. [15] To identify the severity of psychosis in bipolar patients using sparsely labeled voice data, the author introduced a multi-instance learning framework based on deep neural networks. Speech samples were divided into a collection of segments and used as input to deep neural networks for instance-level categorization. Then, using bag-level classification methods, predictions were made for the labels of the speech samples. The most effective method was a multi-instance ensemble learning framework that used a weighted rule.

Tomasik, J. et al. [18] these authors research demonstrates the feasibility of using an empirically grounded algorithm to identify BD in newly diagnosed major depressive disorder (MDD) patients. The results could also apply to other clinically significant groups.

Vasu, V., & Indiramma, M. [19] Datasets, together with the right classifier approach, were the most important factors in analyzing mental health issues, it has been concluded. Even while Electrocardiogram (ECG) data was more reliable than other sources, it still cannot rely on Heart rate variability (HRV) mode. Since the brain and retina have a similar anatomical structure, a retinal picture may be used to more accurately predict bipolar disorder.

Wollenhaupt-Aguiar, B. et al. [20] these authors research have shown that the combination of machine learning methods with peripheral biomarkers may be a powerful diagnostic tool for separating BD patients from those with MDD and HC

3. Materials and Methods

3.1 Dataset

The dataset collected from https://www.kaggle.com/datasets/arashnic/the-depression-dataset contains data related to depression, not specifically bipolar disorder. In bipolar disorder, a person's mood fluctuates wildly, alternating between manic and depressive episodes. It is important to note that the dataset you provided may not specifically focus on bipolar disorder, but rather on depression as a broader category.

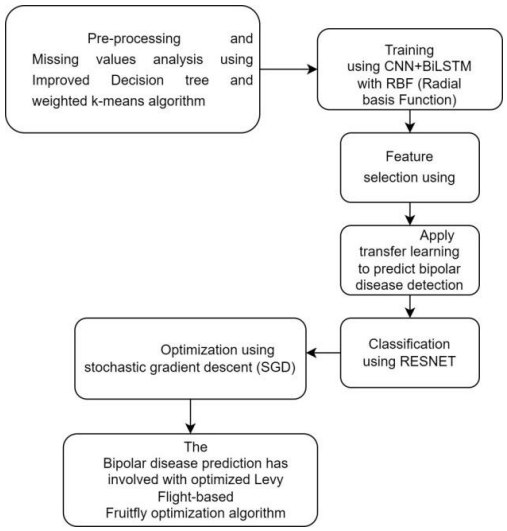

Fig 1: Proposed Architecture

3.2 Pre-processing and Missing values analysis using Improved Decision tree and weighted k-means algorithm

3.2.1 Decision Tree (DT)

If you're having trouble classifying data, DT is the go-to ML approach since it's the most well-known, straightforward, and straightforward. DT is the foundational algorithm from which many sophisticated methods are built. Bagging, RF, and boosting are three prevalent forms of advanced techniques. In the DT world, C5, ID3, and CART 4.5 are all well-known. Typically in DT, we have one node stand in for the input identifier (X), and from its fragments we may infer that X is a numerical value. Forecasting relies heavily on the unique identifier (y) produced by terminal type nodes at the DT's leaves. In most cases, the root type node is selected using DT. The nodes should be divided only after their IG values or entropy values have been measured. Select the node with the highest IG and lowest entropy. Separate the node and proceed as before. This process is continued until there is no possible division or the entropy value has reached its minimum. The amount of uncertainty or unpredictability in data may be measured most accurately using entropy.

$$H((s) = \sum_{x \in X} -p(x)\, log_2 p(x) \text{ ----- (1)}$$

$$IG(A,S) = H(s) - \sum_{t \in T} p(t)\, H(t) \text{ ----- (2)}$$

Here, an entropy-based tree structure has been implemented. The high-depth tree yields more precise results. Predicting a student's future performance based on the facts available now is very useful. Overfitting may be prevented with the use of the DT pruning procedure or branch cropping techniques.

3.2.2 Weighted k-means

Weighted k-means is a variant on the classic k-means clustering technique that uses variable values for each data point at various clustering phases. The conventional k-means method treats all data points equally regardless of their significance. However, in reality, certain details may be more important than others for distinguishing across groups. It is possible to predetermine the weights, or to use a learning method to be taught them. During the clustering process, the algorithm gives more weight to the most important data and less to the less important.

By reassigning data points according to their weighted distance to the nearest cluster center, the weighted k-means technique iteratively updates cluster centers. After a certain amount of cycles, the algorithm stops working if there has been little change to the cluster centers. The weighted k-means method has many benefits over the original k-means algorithm, including improved resilience against outliers, higher quality clusters, and quicker convergence. This method has seen extensive usage in a variety of fields, including image segmentation, data mining, and bioinformatics.

Put a to indicate the break. The difficulty of deciding may be reduced if a cost function were to be developed to evaluate the success of clustering in relation to individual subsets of the dataset. In this case, each number represents a property of each item (gene). So, the amount of characteristics an object possesses may be represented as a row vector of real numbers of length d. Let's assume that all items have the same amount of features and there are no missing values in the collection. Take the set to represent n distinct elements $(x_i, I = 1, n)$. For the sake of brevity, we shall abbreviate the jth property of x_i asx_{ij}. $X = (x_{ij})$ Represents a D attribute matrix for an object set. The cost function for a weighted k-means clustering approach may be expressed as:

$$j_G(\Delta) = \sum_{k=1}^{k} \sum_{x_i \in D_k} (x_i - m_k) G (x_i - m_k) \text{ -------- (3)}$$

$$m_k = \frac{1}{n_k} \sum_{x_i \in D_k} x_i \text{ ------- (4)}$$

Where n_k and m_k are the means and the number of items in D_k, respectively, G is a positively skewed symmetrical weighted matrix. For a weighted k-means method to work, it must find a partition defined by * and a symmetric positive matrix G* satisfying Equation (5), where

$$j_g(\Delta^*) = \min_{\Delta} \{ j_g(\Delta) \} \text{ ------- (5)}$$

When j_g is computed by multiplying a partition by a weighted matrix G, the resulting value might vary. Thus, it is necessary to normalize the weighted matrix. In this investigation, we assume that G = 1, which means.

$$\big(det(G)\big) = 1 \text{ ------- (6)}$$

Condition (4) is met by virtue of the fact that G = I in (3), and equations (5) and (6) provide the cost function and optimum goal of a typical k-means algorithm.

3.3 Training using CNN with Bi-LSTM with RBF (Radial basis Function)

3.3.1 CNN with Bi-LSTM

It is gratifying to see researchers exploring the possibilities presented by CNN's advancements in computer vision to improve E.C. forecasting and load forecasting. As a result, we have introduced a CNN with Bi-LSTM hybrid network for energy consumption forecasting. If you were to draw a map of this network, CNN would be a major hub. After the CNN layer, Bi-LSTM is used to recognize and interpret the sequence of results. The E.C. variables may be mined for temporal aspects with the help of the CNN network. This allows the Bi-LSTM network to provide more precise

predictions of E.C. (electrical conductivity). There are two secret layers in a CNN system: the convolutional and pooling layers.

We provide a methodology wherein household energy consumption may be predicted using historical training models.

1) Data Preprocessing: We dealt with the large number of missing values in the dataset by replacing them with the mean value for that column. For data that does not fit neatly into a typical scale, we use a min-max scalar with a [0, 1] range, the formula for which is provided (7).

$X_n = \left[\frac{X - X_{min}}{X_{max} - X_{min}}\right]$ ------ (7)

The normalized values of the dataset are X_{max} , X_{min}, and X_n, where X represents a value at a certain time step. It's crucial to undertake data preparation before running any calculations to save time and money.

2) During feature selection, we employed a sliding window strategy to convert our dataset into features and labels. We used the current value as the label and the past values in the GAP column as features. The suggested method use a window size of 60, with the preceding 60 values serving as features for the 61st value (the label).

3) The suggested method makes use of the LSTM, Bi-Bi-LSTM, CNN-LSTM, and CNN-LSTM network architectural models. The network efficiency of a given model shifts with its architecture. Changes to the number of filters, kernel size, and

Algorithm 1 CNN with Bi-LSTM

Input:

- Input data or feature vectors (e.g., images) to the CNN with Bi-LSTM model
- Retrained weights for the CNN layers
- Hyperparameter such as the number of CNN filters, kernel sizes, and pool sizes
- Hyperparameter for the Bi-LSTM layer, such as the number of hidden units and dropout rate
- Target labels or classes (if performing classification)

Algorithm Steps:

1. Input data is passed through the CNN layers to extract high-level features:
    - Apply convolutional layers with filters, kernel sizes, and activation functions.
    - Perform pooling operations (e.g., max pooling) to down sample the features.

    $X_n = \left[\frac{X - X_{min}}{X_{max} - X_{min}}\right]$
2. The output from the last CNN layer is fed into the Bi-LSTM layer:
    - Convert the 2D feature maps from the CNN into a 1D sequence for input to the Bi-LSTM.
    - Apply a Bidirectional LSTM layer to capture temporal dependencies in the data.
3. Optionally, additional fully connected layers or other types of layers can be added after the Bi-LSTM layer for further processing or dimensionality reduction.

Output:

Predicted output or classification labels for the input data

3.3.2 Radial basis Function

The Lagrange multipliers provide the foundation of Fasshauer's RBF approximation. This subsection will provide a concise overview of the method's salient features.

A restricted quadratic optimization problem is used to develop this RBF approximation. Approximating the provided dataset by function is the objective of this approach:

$f(x) = \sum_{j=1}^{M} c_j \emptyset \left(\left|\left|X - \epsilon_j\right|\right|\right)$ ------ (8)

When each of the M RBFs is linked to a distinct reference point j and given a weight by a suitable coefficient c_j , and the approximation function $f(x)$ is expressed as a sum of these RBFs. Because of this, finding the weight vector $c = (c1, \ldots, cM)$T is essential for minimizing the quadratic form:

$\frac{1}{2} C^T Q_c$ ------ (9)

for any positive definite matrix Q that is M × M symmetric. Thus, the LSE is a good way to characterize the restricted quadratic minimization issue:

$F(c, \gamma) = \frac{1}{2} C^T q_c - \gamma^T (Ac - h)$, ------- (10)

Within the $set\_ = (1, .., N)$ We must determine the minimum of (_) with respect to c and _, where T is the vector of Lagrange multipliers. Thus, the following system may be solved:

$\frac{aF(c,\mu)}{ac} = q_c - A^T \mu = 0$ ------- (11)

$\frac{aF(c,\mu)}{a\mu} = Ac - h = 0$ ------- (12)

or, in matrix form:

$$\begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} c \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \\ h \end{pmatrix}$$ ------- (13)

Note that we want to minimize M such that the computing cost of the estimated value f(x) is minimized to the maximum extent feasible. Alternatively, one may use RBF interpolation, which relies on the solution of an LSE:

$Ac = h$ ------- (14)

in where A is the system's matrix, c is the variables' column vector, and h is the equations' right-hand side vector. A major drawback of RBF interpolation is the LSE's enormous and often unconditioned matrix. Also, while dealing with an oversampled dataset or with the intention of reducing it, we want to minimize the issue at hand by cutting down on the amount of weights and basis functions employed, all while maintaining a high level of accuracy in the estimated solution. The reduction is a part of the method known as RBF approximation. The following is a detailed description of the procedure that was recently introduced in.

This grid is not guaranteed to include all of these points of reference. Their positioning should be such that it mirrors the surface as closely as possible. The accuracy of the estimated values from the underlying data is enhanced by strategically placing the reference points. Placement along characteristics like break lines, for instance, improves approximation outcomes when a landscape is to be approximated. In a set of N reference points, the number of additions to the set _j is M. Finding the distance between two points in the expanded dataset, xi and _j, is the foundation of the RBF approximation.

The approximated value can be determined similarly as for interpolation

$f(x) = \sum_{j=1}^{M} c_j \emptyset(r_j) = \sum_{j=1}^{M} c_j \emptyset\left(\left||x - \varepsilon_j|\right|\right)$ ------ (15)

When each of the M RBFs is linked to a distinct reference point j and given a weight by a suitable coefficient c_j, and the approximation function $f(x)$ is expressed as a sum of these RBFs.

It can be seen that we get an over determined LSE for the given dataset:

$h_i = f(x_i) = \sum_{j=1}^{M} c_j \emptyset\left(\left||x_i - \varepsilon_j|\right|\right) = \sum_{j=1}^{M} c_j \emptyset_{i,j} \quad i = 1, \dots, N.$ ------ (16)

3.4 Feature selection using Deep Neural Networks

The fundamental idea behind neural networks is that each neuron processes data by applying a weighted average to the values sent into it. When synapses scale values up and are then added together in a neuron, the result is a weighted sum. Although the computation linked to a network of neurons would be a straightforward linear algebraic operation, the neuron in question does more than just output the weighted sum. As a result of the combined inputs, a functional event takes place within the cell. It seems that the neuron is nonlinearly functional, since it is activated to produce an output when the inputs exceed a certain threshold. Therefore, neural networks mimic this process by applying a non-linear function to the input values' weighted sum.

The network's input layer neurons take in information and pass it on to the hidden neurons in its middle layer. The final outputs of the network are sent to the user through the output layer, which receives the weighted sums from one or more hidden layers.

$$u_{k,l} = \mathrm{ReLU}(u_{k,l}) = \begin{cases} u_{k,l} & \text{if } u_{k,l} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$ ------ (17)

For any k and l with k and $u_{k,l}$, we have every node linked to every node in the previous layer using pretrained parameters, with the exception of inputs:

$u_{k,l} = b_{k,l} + \sum_{1 \leq h \leq s_{k-1}} w_{k-1}, h, l\, u_{k-1}, h$ ------ (18)

The so-called bias of node $u_{k,l}$ is denoted by $b_{k,l}$. We point out that this formulation may be used to define both fully-connected functions. Then, the DNN assigns each input a label, the node index in the output layer with the most significant value: We may write this as label = argmax11 $\leq$ h $\leq s_{k-1}$ denotes the collection of labels.

3.5 Classification using RESNET

n this case, we are interested in ResNet in its entirety, pre-activation version included. We just take identity mappings into account for shortcut connections. For simplicity's sake, we will not include the raw input or the most recent linear classifier. Straight after the raw input, you could see a stem block or a few conventional convolution layers. For the sake of brevity, we also leave them out.

Unit i's trainable non-linear mappings, commonly known as Block i, use the value of yi−1 as input. Unit i's output is specified recursively as:

$y_i = f_i(y_{i-1}, w_i) + y_{i-1}$ ------ (19)

Where w_i are the parameters that may be trained and f_i is typically a stack of two or three convolution stages.

Specifically, it uses a label encoder to normalize the input. Labels lacking numbers are changed to

corresponding numerical ones. Using TF-IDF, word counts, or the frequency with which each word occurs in the text, the Tokenize transforms each word into an integer sequence or vector with a binary coefficient.

Tf, or token frequency, is the total number of times a certain token appears in a given content record. The percentage of times this token appears in the content record compared to the total number of tokens in equation 1.

$$\mathrm{tf_{ij}} = \frac{\mathrm{n_{ij}}}{\sum_k \mathrm{n_{ij}}} \text{------ (20)}$$

To determine how often out-of-the-blue tokens show up in archived records, statisticians use the Inverse Data Frequency (IDF) statistic. It's more likely that the tokens that only occur seldom in the record document (i.e.2)

$$\mathrm{df(w)} = \log(\frac{\mathrm{N}}{\mathrm{df_i}}) \text{ ------- (21)}$$

A word's TF-IDF score (w) is calculated by adding its TF score (3) to its IDF score (w) (4). Specifically, I refer to the following equation 3,

$$\mathrm{W_{i,j}} = \mathrm{tf_{i,j}} \times \log(\frac{\mathrm{N}}{\mathrm{df_i}}) \text{ ------- (22)}$$

$\mathrm{tf_{i,j}}$= counting the occurrences of I in j

$\mathrm{df_i}$= records where I is the id value

N = the whole count of files

Tokens are converted to word sequences using the text to sequence tool, which is then used to train the model.

Algorithm 2 RESNET

Input:

- Input tensor $y_i - 1$ representing the input to the residual unit i
- Trainable parameters w_i for the non-linear mappings in Block i

Steps:

1. Take the input tensor$y_i - 1$.
2. Apply the trainable non-linear mappings (Block$i$) to the input tensor$y_i - 1$, using the trainable parametersw_i.
   $y_i = f_i(y_{i-1}, w_i) + y_{i-1}$
3. Compute the output of Block i as$f_i(y_{i-1}, w_i)$.
   $\mathrm{tf_{ij}} = \frac{\mathrm{n_{ij}}}{\sum_k \mathrm{n_{ij}}}$
4. Add the input tensor yi-1 to the output of Block$i$.
   $\mathrm{W_{i,j}} = \mathrm{tf_{i,j}} \times \log(\frac{\mathrm{N}}{\mathrm{df_i}})$
5. Return the output tensory_i.

Output:

Output tensor y_i representing the output of the residual unit i

Model training may be done for any given epoch count and batch size. model concordance (seq mat, train, batch size, epochs, validation split). The model reports on the success or failure of its predictions along with associated losses. Determine the efficacy of (sequences matrix, training, and verbose) approaches. Use the model to anticipate the result. Estimate (try, batch size, detailed).

ResNet Structure Optimization

A unique kind of directed graph may be used to describe the ResNet architecture. If we have a ResNet B=(X, T) with a finite number of neurons, X = xi, and a collection of weights, T=Tk | (xi, xj), then we may say that (xi, xj) is a tensor between xi and xj.

Connecting two neurons in the network X. Assuming we have a dataset of time series N = Ni, we can define the loss function of TSC as follows in equation 4:

$$\mathrm{D(N, B)} \text{ ------ (23)}$$

(X, T•U), where U=uk, uk(0, 1) is a transition matrix, yields an improved ResNet structure B'. In order to optimise the structure, one may write in equation 5:

$$\min \|\mathrm{D(N, X, T)} - \mathrm{D(N, X, T \cdot U)}\| + \lambda\Omega(\mathrm{U}) \text{ ------- (24)}$$

The difference between the original B structure and the improved H structure is denoted by ||•||. Transition matrix U has a penalty function denoted by B', (U). The formal formulation of the ResNet structure optimization goal function is as follows in equation 6:

$$\mathrm{argmin U} \|\mathrm{D(N, X, T)} - \mathrm{D(N, X, T \cdot U)}\| + \lambda\Omega(\mathrm{U}) \text{ ------- (25)}$$

Consistent with the preceding discussion, the computational complexity of this structural optimization is O(2n), placing it in the class of problems known as NP-hard. It is well-known that NP-hard issues are where GAs attracted the most interest. In this way, GA may be used to improve the design in question. Specifically, Algorithm 1 demonstrates how the structure of a ResNet network may be optimised.

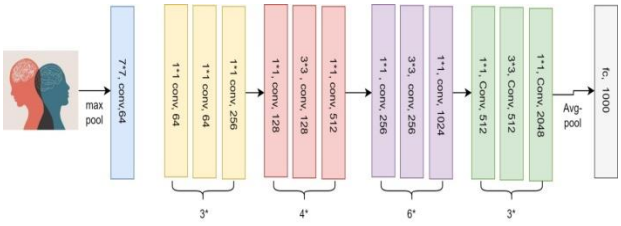

Fig 2: RESNET architecture

3.6 Optimization using stochastic gradient descent (SGD)

One typical approach to addressing machine learning problems is to minimize a loss function, which can be

represented as an expectation over any underlying distribution x ∼ D.

$$\min_{w \in R^d} E_{x \sim D} [\emptyset(x, w)] \text{ ------- (26)}$$

Take the following example: x might represent a picture and D the distribution of images seen in nature. A possible indication of w's ability to explain x is φ(x,w). A few samples (also termed examples) x_1,...,x_n allow us to get the expectation. Each xi may represent a different picture, for instance. Solving the following is a logical approach as it is the only method to access the distribution D via the variables x_1,...,x_n:

$$\min_{w \in R^d} \frac{1}{n} \sum_{i=1}^{n} \emptyset\,(x_i, w) \text{ ----- (27)}$$

After establishing the function, the stochastic gradient for every given step may be calculated by randomly selecting an integer between 1 and n and applying. The whole gradient of the function may be calculated in (26), as you can see. Nevertheless, it takes Ω(n) time to review all the cases. To sum up, computational efficiency is the main selling point of SGD when it comes to machine learning applications. Finite sum issues are the technical terms used to describe this kind of optimization issue.

Understanding convergence rates of finite sum problems is of primary relevance to us since this is the form that most machine learning problems can be expressed in. It turns out, however, that studying how SGD handles easier issues might teach us a lot about intuition. Take the case when we want to minimize f(w) and we have as an example. In the sections that follow, we will also provide findings for these intriguing settings.

3.7 The Bipolar disease prediction has involved with optimized Levy Flight-based Fruitfly optimization algorithm.

3.7.1 Optimized Levy Flight-based Fruitfly optimization algorithm

Similar to the way fireflies flash, firefly algorithm (FA) is based on a natural phenomenon. A value that can be found by solving the goal function is denoted by a firefly in the population matrix. As the number of choice variables in the criteria function increases, so does the number of dimensions in a firefly. The firefly uses the exploitation component to guide it through the search space of the target function, while the random component may be tuned to suit a variety of purposes. The firefly's degree of fitness is reflected in the brightness of its glow. In order to bring the less bright firefly closer to the more bright ones, an updating algorithm is used. The intensity of a firefly's light, Ir, is proportional to its distance, R, according to the inverse-square law. In light of the following:

$$I_r = \frac{I_s}{R^2} \text{ ----- (28)}$$

The absorption coefficient of the medium affects the brightness of the firefly, and it is represented as:

$$I = I_o e^{-rR} \text{ ----- (29)}$$

where Io represents the intensity at the origin (R = 0). The preceding two relations may be coupled as follows to get rid of the singularity for (12) at R = 0:

$$I = I_o e^{-rR^2} \text{ ------ (30)}$$

Every firefly's intensity is proportional to its allure, hence the (14) may be rewritten as follows:

The suggested enhancement to classic FA enhances its search capabilities by including both social and cognitive components. The major goal is to influence firefly F_a migration towards not only the local firefly F_b, but also the global firefly Fg. The capacity of c1 and c2 to govern individual steps in the altered FA's life cycle will be discussed in the next section.

Algorithm 3: optimized Levy Flight-based Fruit fly optimization algorithm

Input:

- Population matrix with fireflies and their corresponding positions in the search space
- Objective/goal function to be optimized
- Parameters: absorption coefficient (r), attractiveness coefficient (beta), maximum iterations, and population size

Steps:

1. Initialize the population matrix with random firefly positions in the search space.
   $I_r = \frac{I_s}{R^2}$
2. Evaluate the fitness of each firefly based on the objective/goal function.
   $I = I_o e^{-rR}$
3. Iterate through the maximum number of iterations or until convergence is reached.
   $I = I_o e^{-rR^2}$

Output:

Updated population matrix with fireflies after each iteration, representing their new positions in the search space

4. Results and Discussion

The results provide valuable information for decision-making and further analysis. They enable us to identify the strengths and weaknesses of each model and make informed choices about which model to use for the specific classification task at hand.

In this case, the RESNET model demonstrated the highest performance across all evaluated metrics, achieving the highest values in Accuracy, Precision,

Recall, and F1-Score. This indicates that the RESNET model outperformed the CNN and DNN models in terms of overall predictive accuracy, precision in positive predictions, recall of actual positive cases, and a balanced measure of precision and recall.

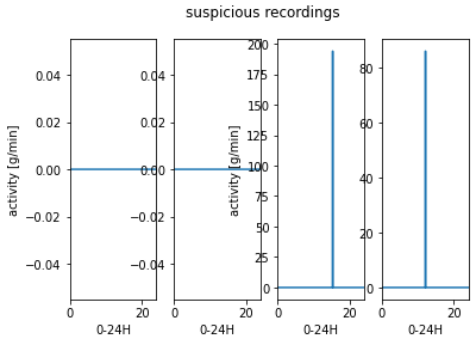

Fig 3: suspicious recordings

The figure 3 shows suspicious recordings

Table 1: K and P value comparison table

	K-value	P-value
day-wise mean	11.4165	3.4583
sbubject-wise mean	1.8257	0.0678
day-wise max	8.1131	4.9326
subject-wise max	1.1688	0.2424
subject-wise Acrophase	1.5868	0.1125
subject-wise Cycle Amplitude	0.4948	0.6207

The K-value and P-value, among other metrics, are shown in table 1 under several headings. The K-value stands for a certain attribute or parameter, whereas the P-value denotes the likelihood or statistical significance of the relevant measurement. To begin, there is a consistent trend or pattern in the data over multiple days, as shown by the average K-value of 11.4165 in the "Day-wise Mean" category. The observed mean value is very unlikely to be the result of chance, as shown by the moderate degree of statistical significance suggested by the related P-value of 3.4583. The average K-value in the "Subject-wise Mean" category is 1.8257, which is much lower. As a result, it's possible that the parameter being tested varies among individuals. With a matching P-value of 0.0678, the degree of statistical significance is low, suggesting that the observed mean value might be more likely to be due to chance. For details on the highest K-values recorded, check out the "Day-wise Max" and "Subject-wise Max" sections. The subject-wise maximum is lower at 1.1688, whereas the day-wise maximum is recorded at 8.1131. There is a range of statistical significance for these highest values, as shown by the related P-values of 4.9326 and 0.2424, respectively. Last but not least, we have "Subject-wise Acrophase" and "Subject-wise Cycle Amplitude," which stand for certain subject-related traits or dimensions. Certain patterns or qualities are indicated by the values of 1.5868 and 0.4948, respectively. The statistical importance of these traits may be shown from the matching P-values of 0.1125 and 0.6207. Summarised in the table are the measurements denoted by the K-value and the corresponding P-values. The relevance of these numbers may alter across various domains of research or applications, so the interpretation of the data is domain and context dependent.

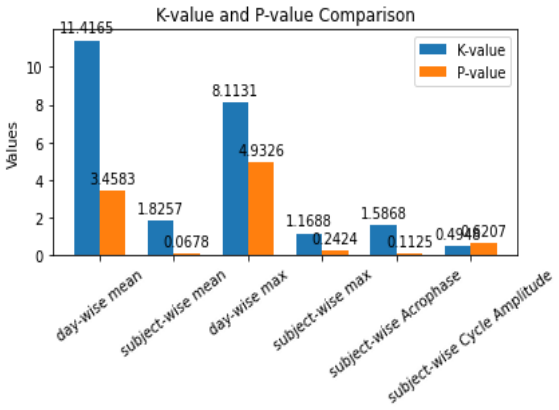

Fig 4: K-value and P-value comparison

Various metrics' K-values and P-values are compared in Figure 4. On one side, we have the metrics under scrutiny, and on the other, we have the K-value and P-value numbers. An informative quantitative indicator for a particular parameter or attribute, the K-value sheds light on the extent to which that metric is significant. Every measure in the research or analysis has a numerical value that it reflects.

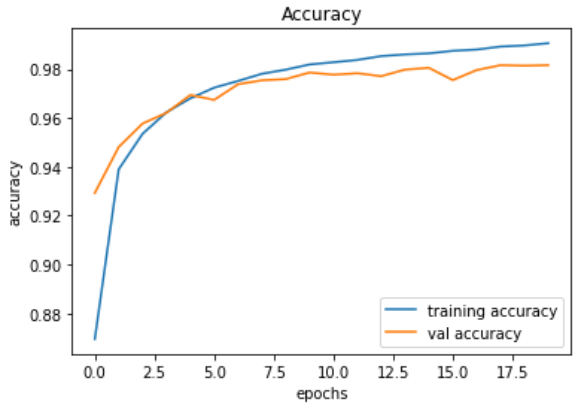

Fig 5: training accuracy

Figure 5 displays the training accuracy of a model across a large number of epochs. The number of complete iterations of the training dataset that the model goes through during training is represented by the epochs, which are depicted on the x-axis. The model's accuracy values at each epoch are shown on the y-axis. While training a model, its performance on the training dataset is evaluated using the accuracy measure. This metric quantifies the percentage of training dataset samples that were correctly classified as a percentage of the total training dataset samples.

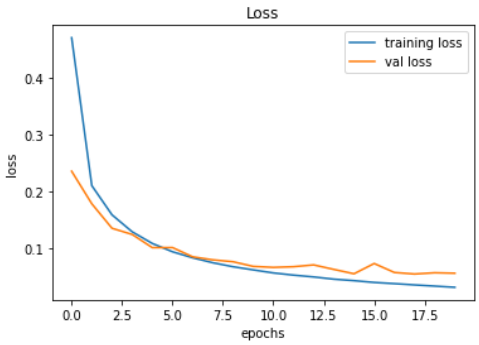

Fig 6: training loss

As seen in Figure 6, a model's training loss across many epochs is shown. As the model is being trained, the x-axis shows the number of epochs, which are the total number of iterations through the training dataset. For each time period, the y-axis shows the corresponding loss values that the model calculated. Using the loss function, we can measure the extent to which the model's projected outputs deviate from the actual target values when training the model. The loss is a numerical number that indicates the degree to which the actual values differ from the predicted values. It serves as a gauge for the model's performance in fitting the training data.

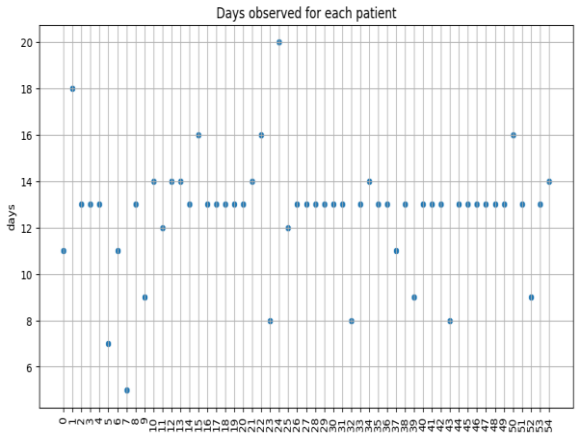

Fig 7: days observed for each patient

Figure 7 shows the data visually, the "days observed for each patient." The count is given on the x-axis, while the number of days observed is indicated on the y-axis. This graphic provides insight into the duration of observation for each topic in the dataset. The data for each patient was recorded or tracked for a certain duration, as shown on the y-axis, while the total number of patients is displayed on the x-axis. The graphic allows one to see the variability in the duration of patient monitoring. Potential applications include identifying patterns in data, such as the distribution of patients across time, outliers, data gaps, and the range of observation periods.

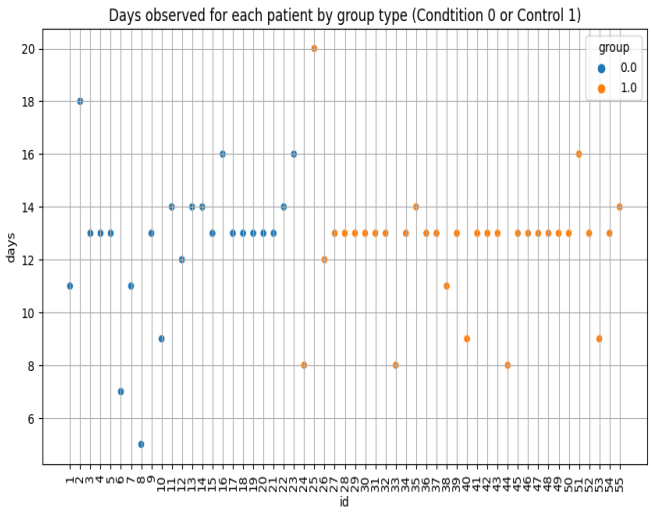

Fig 8: days observed for each patient by group type

Figure 8 provides an aesthetically appealing breakdown of the number of days observed for each patient based on the group type. The length of time it takes to collect data from different patient groups may be better understood by comparing the observation times.

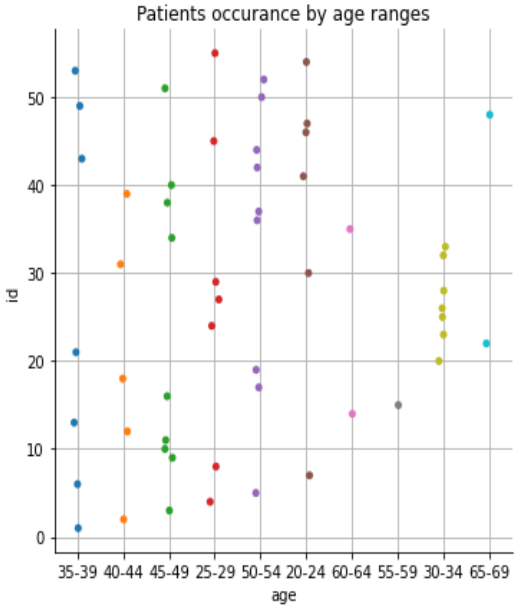

Fig 9: patient occurrence by age range

We can see the distribution of patients across different age groups in Figure 9, which displays the frequency of patients by age range. In addition to providing insight into patient demographics by age group, it also provides

a visual representation of the overall patient count across all age categories.

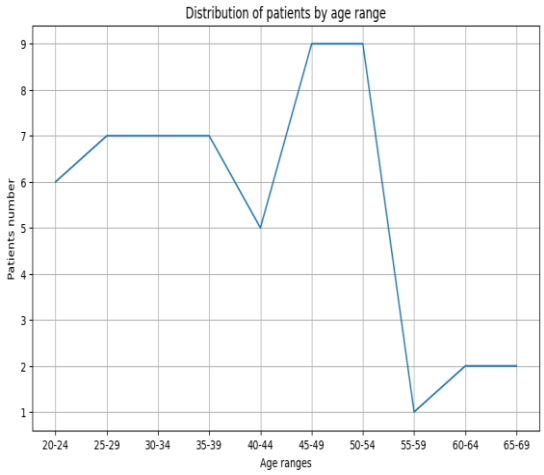

Fig 10: Distribution of patients by age range

Figure 10 shows that the number of patients in each age group clearly illustrates the distribution of patients by age range. By doing so, we may learn more about the distribution and frequency of patients in the dataset across various age groups.

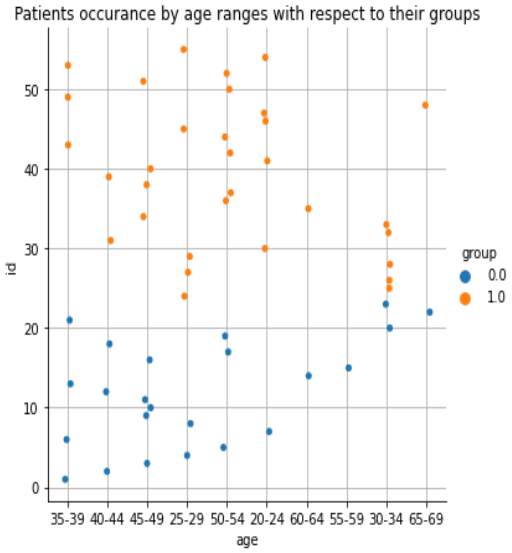

Fig 11: patient's occurance by age range with respect to their groups

With group membership in mind, Figure 11 provides a graphical depiction of patient incidence by age range. This method enables researchers to examine the demographic makeup of the patient population and age-related trends by comparing the distribution of patients across various age intervals within each category.

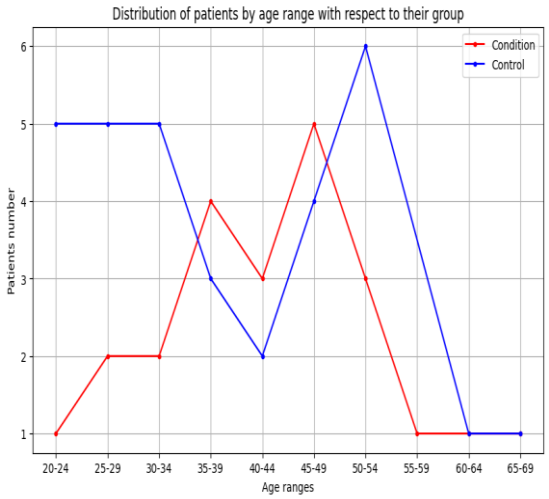

Fig 12: Distribution of patients by age range with respect to their groups

Patients' age distributions, adjusted for group membership, are shown in Figure 12. An easy way to compare patient distribution across various age intervals while considering their respective groups is by looking at the visual depiction of the number of patients in each age group inside each group.

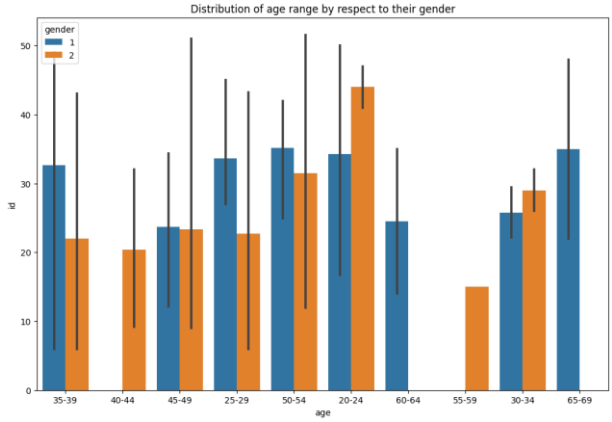

Fig 13: distribution of age range by respect to their gender

Here is a graphic depiction of the gender distribution of age groups in Figure 13. Insights about the demographic make-up and age-related traits of each gender category may be gleaned from this examination of gender-based age distributions.

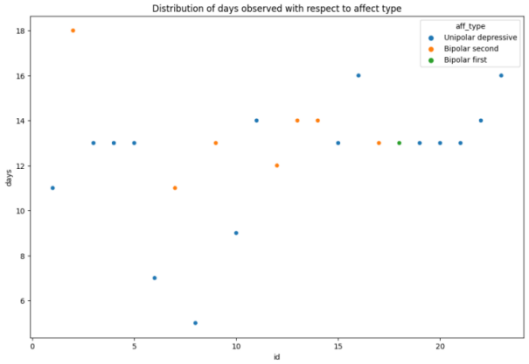

Fig 14: distribution of days observed with respect to affect type

In Figure 14, we can see the breakdown of the days observed according to the different types of influence. The x-axis displays the patient IDs, while the y-axis displays the total number of days noticed. Here we may see the distribution of effect types by number of days. On the x-axis, you can see the patient IDs, which stand for distinct people in the dataset. The y-axis shows the total number of days that each patient was monitored.

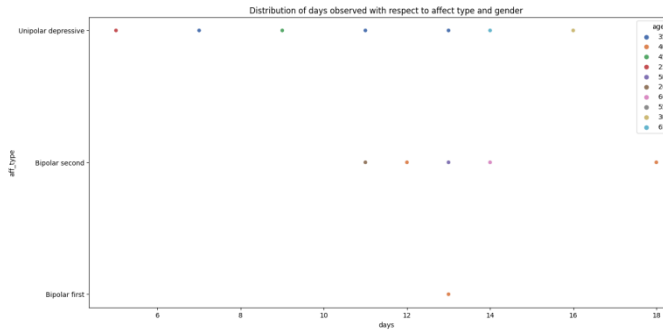

Fig 15: distribution of days observed with respect to affect type and gender

Figure 15 visually represents the distribution of days observed with respect to affect type and gender. The x-axis represents the number of days observed, while the y-axis represents the affect types. This figure provides insights into the temporal distribution of observations across different affect types and genders. The affect types are displayed on the y-axis, representing various emotional states or affective categories.

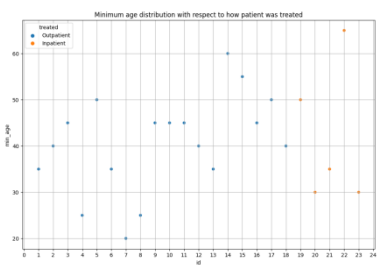

Fig 16: minimum age distribution with respect to how patient was treated.

Patients' minimum ages were distributed according to their treatments, as shown in Figure 16. On one side of the graph we see the patient IDs, while on the other we see the minimum age. The distribution of minimum age values for patients is shown in this graphic, which illustrates the influence of different treatment modalities. The x-axis displays the patient IDs, which represent unique individuals within the collection. The y-axis displays the minimum age that has been reported for each case.

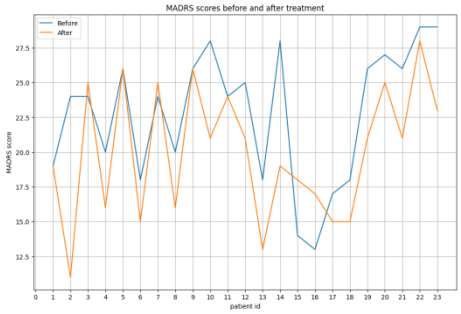

Fig 17: MADRS scores before and after treatment

Before and after treatment, the scores on the Montgomery-Åsberg Depression Rating Scale (MADRS) are shown in Figure 17. The patient IDs are on the left and the MADRS scores are on the right. This picture shows the pre- and post-treatment changes in the MADRS scores of each patient. The x-axis displays the patient IDs, which represent unique individuals within the collection. The depression severity scores (MADRS) are shown on the y-axis.

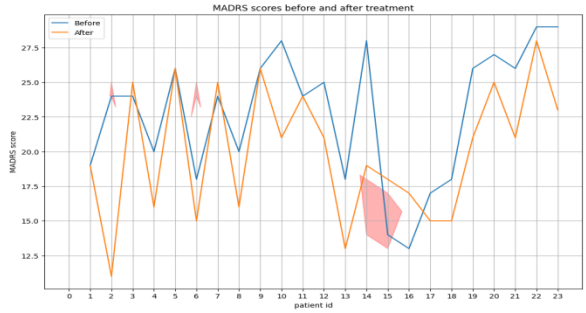

Fig 18: MADRS scores before and after treatment

Figure 18 visually displays the scores on the Montgomery-Åsberg Depression Rating Scale (MADRS) both before to and during treatment. The patient IDs are on the left and the MADRS scores are on the right. This picture shows the pre- and post-treatment changes in the MADRS scores of each patient. The x-axis displays the patient IDs, which represent unique individuals within the collection. The depression severity scores (MADRS) are shown on the y-axis.

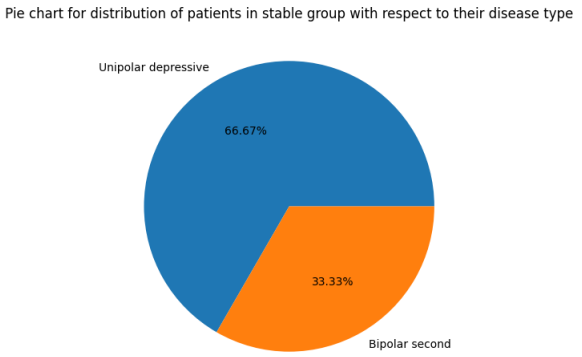

Fig 19: pie chart

Figure 19 represents the distribution of patients with bipolar disorder and unipolar depressive disorder using a pie chart. The chart visually depicts the proportions of each disorder category. In the pie chart, the category "bipolar second" occupies 33.33% of the total chart area, while the category "unipolar depressive" occupies 66.67% of the total chart area.

Table 2: performance metrics comparison

Performance metrics			
	CNN	**DNN**	**RESNET(Proposed)**
Accuracy	0.9532	0.9632	0.9806
Precision	0.9489	0.9603	0.9806
Recall	0.9132	0.9354	0.9803
F1-Score	0.9356	0.9532	0.9804

The table 2 shows for three different models, CNN, DNN, and RESNET, are presented in the table. The metrics evaluated are Accuracy, Precision, Recall, and F1-Score.

For the CNN model, the accuracy achieved is 0.9532, for the DNN model it is 0.9632, and for the RESNET model, it is the highest at 0.9806. A higher accuracy value indicates a better-performing model in making correct predictions.

Precision: The accuracy of a model's predictions is defined as the ratio of its genuine positive predictions to its total number of positive forecasts. The CNN model achieves a precision of 0.9489, the DNN model achieves 0.9603, and the RESNET model achieves the highest precision of 0.9806. Higher precision values indicate a lower rate of false positive predictions.

Recall: A model's recall—also called its sensitivity or true positive rate—is a measure of how well it detected real positive instances. In terms of recall, the CNN model gets 0.9132, the DNN model gets 0.9354, and the RESNET model has the best recall at 0.9803. An improved accuracy in predicting unfavorable outcomes is shown by a higher recall value.

F1-Score: The F1-Score gives a fair evaluation of the model's efficacy as it is the harmonic mean of the recall and accuracy. With an F1-Score of 0.9356, the CNN model outperforms the DNN model by a wide margin, while the RESNET model takes the cake with an F1-Score of 0.9804.

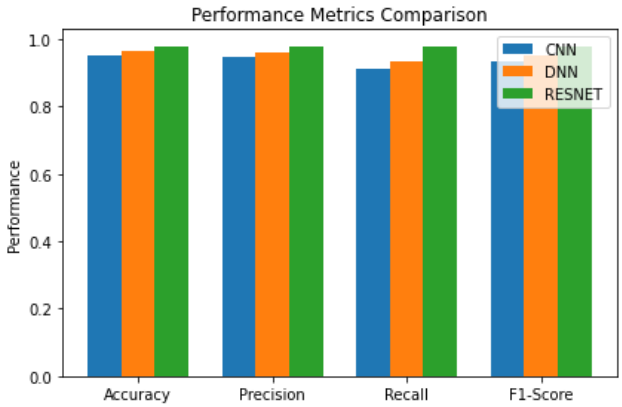

Fig 20: performance metrics comparison

The figure 20 shows performance metrics comparison the x axis shows metrics and the y axis shows performance.

5. Conclusion

We developed a unique transfer learning strategy for bipolar disorder diagnosis using several modules to improve prediction accuracy and efficiency. Our technique includes deep learning architectures, feature selection, transfer learning, RESNET classification, stochastic gradient descent optimization, and an optimized Levy Flight-based Fruitfly optimization algorithm. We use deep learning to extract meaningful features from raw input data using a CNN with BiLSTM and RBF. The DNN for feature selection eliminates duplicate and extraneous data to optimise feature representation. We use transfer learning to a pre-trained model to diagnose bipolar disorder. The trained representations improve prediction performance even with minimal labelled data. The classification module uses the RESNET architecture, which excels in picture classification, to identify complex bipolar illness patterns. Bipolar illness prediction improves. Optimizing model parameters is key. SGD repeatedly updates parameters based on subsets of training data for quicker convergence and better accuracy. An improved Levy Flight-based Fruitfly optimization technique fine-tunes model parameters. This method optimizes learning rate, batch size, and regularization to diagnose bipolar disorder effectively. Our transfer learning strategy

improves bipolar disorder identification by merging various modules and approaches. Early identification improves treatment planning, intervention, and long-term care for bipolar illness patients. Expanding the dataset, adding multi-modal data sources, and researching advanced deep learning architectures may improve the model's prediction power. For the suggested strategy to be practical and reliable, comprehensive validation studies in clinical settings and with healthcare experts are needed.

Reference

[1] Antosik-Wójcińska, A. Z., Dominiak, M., Chojnacka, M., Kaczmarek-Majer, K., Opara, K. R., Radziszewska, W., … Święcicki, Ł. (2020). Smartphone as a monitoring tool for bipolar disorder: a systematic review including data analysis, machine learning algorithms and predictive modelling. International Journal of Medical Informatics, 104131. doi:10.1016/j.ijmedinf.2020.104131

[2] Batchvarov, V. N., & Behr, E. R. (2015). Computed bipolar precordial leads for improved P wave detection. Journal of Electrocardiology, 48(2), 188–189. doi:10.1016/j.jelectrocard.2014.12.009

[3] Constantinou, L., Kyriacou, P. A., & Triantis, I. F. (2017). Towards an optimized tetrapolar electrical impedance lithium detection probe for bipolar disorder: A simulation study. 2017 IEEE SENSORS. doi:10.1109/icsens.2017.8234225

[4] Daus, H., Kislicyn, N., Heuer, S., & Backenstrass, M. (2018). Disease management apps and technical assistance systems for bipolar disorder: Investigating the patients´ point of view. Journal of Affective Disorders, 229, 351–357. doi:10.1016/j.jad.2017.12.059

[5] Delvecchio, G., Pigoni, A., Altamura, A. C., & Brambilla, P. (2018). Cognitive and neural basis of hypomania: Perspectives for early detection of bipolar disorder. Bipolar Disorder Vulnerability, 195–227. doi:10.1016/b978-0-12-812347-8.00010-5

[6] Ding, S.-N., Wang, X.-Y., & Lu, W.-X. (2019). Switches-controlled bipolar electrode electrochemiluminescence arrays for high-throughput detection of cancer biomarkers. Journal of Electroanalytical Chemistry, 844, 99–104. doi:10.1016/j.jelechem.2019.05.021

[7] Fitriati, D., Maspiyanti, F., & Devianty, F. A. (2019). Early Detection Application of Bipolar Disorders Using Backpropagation Algorithm. 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). doi:10.23919/eecsi48112.2019.8977102

[8] Jadhav, R., Chellwani, V., Deshmukh, S., & Sachdev, H. (2019). Mental Disorder Detection : Bipolar Disorder Scrutinization Using Machine Learning. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). doi:10.1109/confluence.2019.8776913

[9] Kessing, L. V., Ziersen, S. C., Andersen, P. K., & Vinberg, M. (2021). A nation-wide population-based longitudinal study mapping physical diseases in patients with bipolar disorder and their siblings. Journal of Affective Disorders, 282, 18–25. doi:10.1016/j.jad.2020.12.072

[10] Kibbi, N., Totonchy, M., Suozzi, K. C., Ko, C. J., & Odell, I. D. (2018). A case of subungual tumors of incontinentia pigmenti: A rare manifestation and association with bipolar disease. JAAD Case Reports, 4(7), 737–741. doi:10.1016/j.jdcr.2018.03.018

[11] Li, Z., Li, W., Wei, Y., Gui, G., Zhang, R., Liu, H., … Jiang, Y. (2021). Deep learning based automatic diagnosis of first-episode psychosis, bipolar disorder and healthy controls. Computerized Medical Imaging and Graphics, 89, 101882. doi:10.1016/j.compmedimag.2021.101882

[12] Librenza-Garcia, D., Kotzian, B. J., Yang, J., Mwangi, B., Cao, B., Pereira Lima, L. N., … Passos, I. C. (2017). The impact of machine learning techniques in the study of bipolar disorder: A systematic review. Neuroscience & Biobehavioral Reviews, 80, 538–554. doi:10.1016/j.neubiorev.2017.07.004

[13] Passos, I. C., Ballester, P., Barros, R. C., Librenza-Garcia, D., Mwangi, B., Birmaher, B., … Kapczinski, F. (2019). Machine learning and big data analytics in bipolar disorder: A Position paper from the International Society for Bipolar Disorders (ISBD) Big Data Task Force. Bipolar Disorders. doi:10.1111/bdi.12828

[14] Perez Arribas, I., Goodwin, G. M., Geddes, J. R., Lyons, T., & Saunders, K. E. A. (2018). A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder. Translational Psychiatry, 8(1). doi:10.1038/s41398-018-0334-0

[15] Ren, Z., Han, J., Cummins, N., Kong, Q., Plumbley, M. D., & Schuller, B. W. (2019). Multi-instance Learning for Bipolar Disorder Diagnosis using Weakly Labelled Speech Data. Proceedings of the 9th International Conference on Digital Public Health - DPH2019. doi:10.1145/3357729.3357743

[16] Sawalha, J., Cao, L., Chen, J., Selvitella, A., Liu, Y., Yang, C., … Cao, B. (2020). Individualized identification of first-episode bipolar disorder using machine learning and cognitive tests. Journal of Affective Disorders. doi:10.1016/j.jad.2020.12.046

[17] Sonkurt, H. O., Altınöz, A. E., Çimen, E., Köşger, F., & Öztürk, G. (2020). The role of cognitive functions in the diagnosis of bipolar disorder: A machine learning model. International Journal of Medical Informatics, 104311. doi:10.1016/j.ijmedinf.2020.104311

[18] Tomasik, J., Han, S. Y. S., Barton-Owen, G., Mirea, D.-M., Martin-Key, N. A., Rustogi, N., … Bahn, S. (2021). A machine learning algorithm to differentiate bipolar disorder from major depressive disorder using an online mental health questionnaire and blood biomarker data. Translational Psychiatry, 11(1). doi:10.1038/s41398-020-01181-x

[19] Vasu, V., & Indiramma, M. (2020). A Survey on Bipolar Disorder Classification Methodologies using Machine Learning. 2020 International Conference on Smart Electronics and Communication (ICOSEC). doi:10.1109/icosec49089.2020.9215334

[20] Wollenhaupt-Aguiar, B., Librenza-Garcia, D., Bristot, G., Przybylski, L., Stertz, L., Kubiachi Burque, R., … Kapczinski, F. (2019). Differential biomarker signatures in unipolar and bipolar depression: A machine learning approach. Australian & New Zealand Journal of Psychiatry, 54(4), 393–401. doi:10.1177/0004867419888027