International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Mental Workload Assessment using RNN

Abitha. k[1], Sweatha. k[2], Mrs. Kapilavani. R.k[3]

[1,2,3]Department of Computer Science and Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Tamil Nadu, India.

***Abstract*-**Modern world requires multitasking and the need for sustained vigilance, which result in work related stress and increase the occurrence of human error. So, methods for estimating mental overload of the brain during work completion are needed. Mental workload affects human performance in specific tasks. So, mental workload estimate is essential for maintaining human health and preventing accidents. This issue reaches an average accuracy of 88.9%.But, this model still need a more number of specifications. The proposed system is validated based on EEG dataset using RNN. This study classifies the dataset with different bandwidth patterns. The project proposes a recurrent neural network to explore cross task mental workload estimation. The proposed model reaches satisfactory classification accuracy.

***Keywords*-**Mental workload, recurrent neural network, EEG.

I. INTRODUCTION

Mental workload is defined as cooperation between work requirements and human capacity or resources**.** Workload is the amount of work of an individual has to do. There is a difference between the actual amount of work and the individual's perception of the workload. Workload can also be classified as the amount of work to be done or the difficulty of the work. Mental workload affects human health. Mental workload can decrease human abilities in terms of memory, reaction, and operation. Because of their high mental workload, some jobs like pilots, soldiers, crew, teachers and surgeons may encounter serious effects. The brain is accountable for information processing, decision making and actions on the outside environment. So, mental workload assessment remains an important topic.

An electroencephalogram (EEG) is a test used to calculate the electrical activity in the brain. Brain cells interact with each other through electrical stimuli. An EEG can be used to help detect potential problems associated with this movement.

An EEG tracks and records brain wave patterns. A conductor called electrodes is attached to the scalp of head with wires. The electrodes send signals to a computer that records the results by analysing the electrical impulse in the brain. The electrical impulses in an EEG recording look like spiking with peaks and valleys. These spikes allow doctors to quickly assess whether there are any abnormal patterns. Any irregularities may be appear like seizures or other brain disorders.

Fig .1.demonstrates different frequency patterns in brain waves. They are alpha, beta, gamma, delta, and theta.

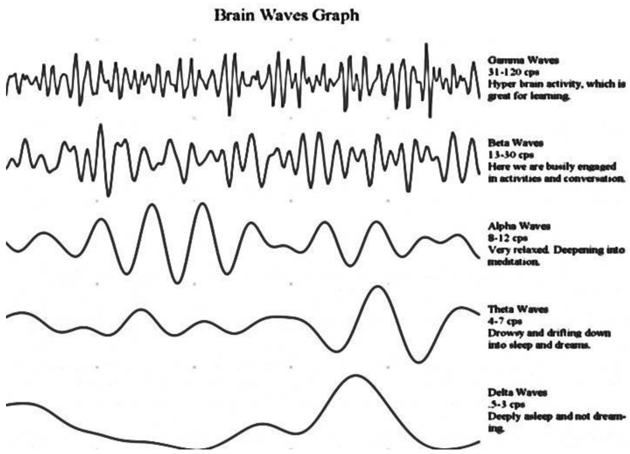

Fig.1. different frequency pattern

1) *Alpha waves*

Alpha brainwaves are dominant during quietly flowing thoughts, and in some meditative states. It is the resting state for the brain. It aid overall mental coordination, calmness, alertness, mind/body integration and learning.

2) *Beta waves*

Beta brainwaves dominate our normal waking state of consciousness when attention is directed towards cognitive tasks and the outside world. A beta frequency becomes stronger as we plan or execute movement of any body part. Beta is a quick activity which present during alert, attentive, engaged in problem solving, judgment, decision making, or focused mental activity. It associated with complex thought, integrating new experiences, high anxiety, or excitement.

3) *Theta waves*

Theta brainwaves occur mostly in sleep but are also in deep rumination. These are associated with learning, memory. In theta, our senses are withdrawn from the external environment and focused on signals originating from within. It is that dusk state which we normally only experience fleetingly as we wake or drift off to sleep. In theta we are in a dream; powerful imagery, insight and information beyond our normal conscious awareness. It's where we hold our 'stuff', our fears, troubled history, and nightmares.

4) *Gamma waves*

Gamma brainwaves are the fastest of brain waves and relate to simultaneous processing of information from different brain areas. Gamma brainwaves pass information rapidly and quietly.

5) *Delta waves*

Delta brainwaves are slow, loud brainwaves. They are associated in deepest meditation and dreamless sleep. Healing and regeneration are stimulated in delta state. So, deep restorative sleep is so essential to the healing process.

II. METHODS

A Neural Network is an information processing model that stimulated by the way biological nervous systems, such as the brain, process information. The key element of this model is the new structure of the information handling system. It is collection of a large number of highly interconnected processing elements (neurons) to solve specific problems. In neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given a new situation of interest and answer "what if" questions.

A **recurrent neural network** (**RNN**) is a type of artificial neural network where nodes are interconnected to form a directed graph along a temporal sequence. This allows it to view temporal dynamic behaviour. Fig.2. represents recurrent neural network. RNNs can use their internal state to process sequences of inputs. This makes them applicable to tasks. The input layer gets the input, apply the activations in hidden layer and then we finally get the output. It has a deeper network, where multiple hidden layers are present.

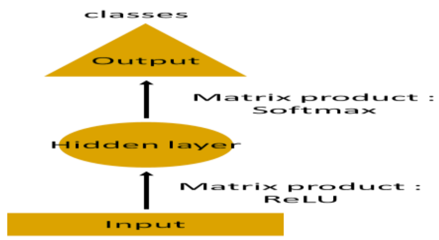

Fig.2.Recurrent neural network

So here, the input layer gets the input, the first hidden layer activations are applied and then these activations are sent to the next hidden layer, and successive activations through the layers to produce the output. Each hidden layer is distinguished by its own weights and biases.

Since each hidden layer has its own weights and activations, they behave independently. To combine these hidden layers together, we shall have the same weights and bias for these hidden layers. All these hidden layers can be rolled in together in a single recurrent layer. At all the time steps weights of the recurrent neuron would be the same since it's a single neuron now. So a recurrent neuron stores the state of a previous input and integrates with the current input thereby conserving some relationship of the current input with the previous input.

RNN applies something called as a recurrence formula to the input vector and also its previous state. In this case, "ht" has nothing proceeding in it. So at the time "ht-1" is supplied to the network, a recurrence formula is applied. These are known as various time steps of the input. So if at time t, the input is "ht", at time t-1, the input was "ht-1". The recurrence formula is applied to ht and ht-1 both. And we get a new state.

The formula for the current state can be written as –

$$\mathbf{h_t = f(h_{t-1}, x_t)}$$

Here, Ht is the new state; ht-1 is the previous state while xt is the current input. We now have a state of the previous input instead of the input itself, because the input neuron would have applied the transformations on our previous input. So each successive input is called as a time step.

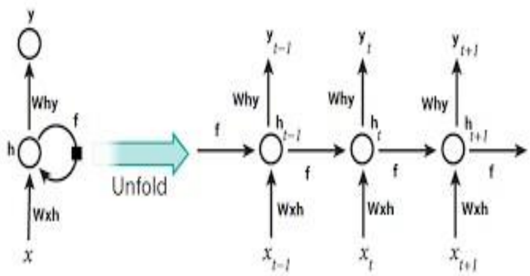

Fig.3. Represents flow of RNN.

In this case we have four inputs to be given to the network, during a recurrence formula, the same function and the same weights are applied to the network at each time step.

Taking the simplest form of a recurrent neural network, let's say that the activation function is tanh, the weight at the recurrent neuron is Whh and the weight at the input neuron is Wxh, we can write the equation for the state at time t as –

$$h_t = \tanh(W_{hh}h_{t-1} + w_{xh}x_t)$$

The Recurrent neuron in this case is just taking the immediate previous state into consideration. For longer sequences the equation can involve multiple such states. Once the final state is calculated we can go on to produce the output.

Now, once the current state is calculated we can calculate the output state as-

$$Y_t = w_{hy}h_t$$

III. EXISTING SYSTEM

The traditional PSD features obtain poor accuracy (less than random probability) for the cross-task problem using a single-hidden-layer back propagation (BP) artificial neural network (ANN). ERS/ERD features also lead to unsatisfactory results for cross-task assessment. Feature selection methods and state-of-the-art classifiers have been used to assess cross-task mental workload, leading to partial improvements in performance. However, the classification accuracy is still limited because of the hand-crafted feature set. In addition to the invalid features, the definitions of the mental workload levels under various types of tasks can also lead to misleading classification results. The labels of the mental workload conditions may be defined subjectively and incorrectly. Based on PSD features, the combination of the support vector machine (SVM) regression model and feature selection method partially solves this problem without comparable classification accuracy. Therefore, hand-crafted EEG features have encountered a bottleneck in solving the cross-task problem.

The cross-task problem appears to comprise many difficulties. The main obstacle originates from the limitations of handcrafted features, which depend on the prior knowledge of experts. The distinguishable information of brain dynamics across different tasks remains complex. Human designed features may neglect the distinguishable information of the raw data. Due to the development of machine learning, deep learning provides a new data-driven approach to solve the EEG classification problem without prior knowledge. In contrast to handcrafted methods, which typically extract features from the temporal and spectral dimensions separately, deep learning methods can learn complex information from multiple dimensions simultaneously. Therefore, researchers have begun using deep learning to learn robust EEG representations.

IV. PROPOSED SYSTEM

In this project, we suggest by RNN Classifier as neural network ensemble that can incorporate different base classifiers into classifier ensembles models for classification problems. This project suggest that the impact of using different base classifiers on classification accuracy of RNN classifier ensemble. Classifier ensembles with five base classifier have used on five medical data sets. These results evaluated and compared choosing different type of decision tree algorithms for base classifier. The reliability of classification for most of datasets and classifier ensembles is increased when we select the appropriate RNN classifier achieves the minimum time required to build models. It is simple to understand and interpret and able to handle both numerical and categorical data, which requires little data preparation, for possible to validate a model using statistical tests, performs well with large datasets. It is robust, which means that performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

V. DESCRIPTION

A. *Pre-processing*

Pre-processing of EEG signal is an essential and important step in any BCI based applications. It helps to remove unwanted artifact from the EEG signal and make it suitable for further processing. It helps to remove unwanted artifacts from the EEG signal and hence improve the signal to noise ratio. The block aids in improving the performance of the system by separating the noise from the actual signal.

B. *Feature extraction*

A feature extraction block helps to get back the most relevant features from the signal. These features will support the decision making mechanism in giving the desired output. EEG signal feature extraction and to show how fast the method used for the signal extraction and how reliable it will be the extracted EEG signal features. Moreover, how these extracted features would express the states of the brain for different mental tasks, and to be able to yield an accurate classification and translation of mental tasks. Therefore the speed and accuracy of the feature extraction stage of EEG signal processing are very crucial, in order not to lose vital information at a reasonable time.

C. *Data cleaning*

Data cleaning is a technique that is applied to discharge the noisy data and correct the discrepancy in data. Data cleaning involves transformations to correct the wrong data. Data cleaning is performed as a data pre-processing step while preparing the data for a data warehouse.

D. *Clustering coefficients of variation (ccv)*

CCV is based on a very easy principle of variance-basis that finds a subset of features useful for optimally balancing the classification model induction between generalization and over fitting. CCV is founded on a basic belief that a good attribute in a training dataset should have its data vary sufficiently wide across a range of values, so that it is significant to characterize a useful prediction model.

E. *RNN classification*

In general, RNN(Recurrent Neural Network) is decision tree based neural network are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. RNN correct for decision trees' habit of over fitting to their training set.

RNN is a statistical algorithm that is used to cluster points of data in functional groups. When the data set is large and/or there are many variables it becomes difficult to cluster the data because not all variables can be taken into account, therefore the algorithm can also give a certain chance that a data point belongs in a certain group.

This is how the clustering takes place.

Of the entire set of data a subset is taken (training set).

The algorithm clusters the data (Stages 1, 2, 3 and 4) in groups and subgroups. If we want to draw lines between the data points in a subgroup, and lines that connect subgroups into group etc. The structure would looks like a tree. This is called a decision tree.

At each division or node in this cluster variables are chosen at random by the program to judge whether data points have a close relationship or not.

The program makes multiple trees a.k.a. a forest. Each tree is different because for each division in a tree, variables are chosen at random.

Then the rest of the EEG dataset (not the training set) is used to predict which tree in the forests makes the best classification of the data points (in the dataset the right classification is known). The tree with the most predictive power is shown as desired output by the algorithm.

F. *Evaluation metrics*

By finding the confusion matrix these are parameters to find it:

1) Accuracy
The proportion of the total number of predictions that was correct.

2) Positive predictive value or precision
The proportion of positive cases that was correctly identified.

3) Negative Predictive Value
The proportion of negative cases that were correctly identified.

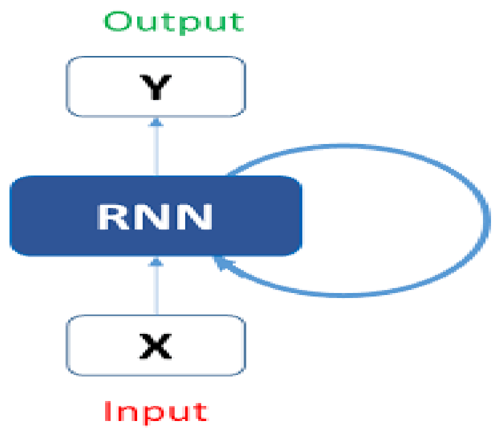

Fig.4.represents input of RNN.

4) Sensitivity or Recall
The proportion of actual positive cases which are correctly identified.

5) Specificity
The proportion of actual negative cases which are correctly identified.

Recurrent neural network are a type of neural network where the output from previous step are feed as input to the current step. An RNN remembers each and every information through time .It is useful in time series prediction only because of the feature to remember previous inputs as well. Recurrent layers having a feedback loop in it. So, it just passes useful information from previous stage to present stage. Finally, evaluate the EEG metrics of the model as normal and abnormal.

VI. IMPLEMENTATION

Implementation is the most critical state in achieving a successful system and giving the user's confidence that the new system is workable and effective. Implementation of a modification application is to replace an existing one. This type of adaptation is relatively easy to handle, provided there are no major changes in the system. Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs qualification, the computer system and its environment is tested to the satisfaction of the user. A simple operating process is included so that the user can understand the different functions clearly and quickly.

Initially at a first step, load the dataset from the online repository. Divide some set of data to train and test the model. Label the dataset as 0 and 1 .Because we use labeled data for supervised learning. In the supervised learning, large numbers of dataset are used to train the neural network. The neural network has number of hidden layer to achieve the high accuracy.

The input layer gets the input, activations are applied in the hidden layer and then we finally get the desired output. It has a deeper network, where multiple hidden layers are present. So here, the input layer receives the input, the first hidden layer activations are applied and then these activations are sent to the next hidden layer, and successive activations through the layers to produce the output. Each hidden layer is distinguished by its own weights and biases.

Since each hidden layer has its own weights and activations, they behave independently. To combine these hidden layers together, we shall have the same weights and bias for these hidden layers. All these hidden layers can be rolled in together in a single recurrent layer. At all the time steps weights of the recurrent neuron would be the same since it's a single neuron now. So a recurrent neuron stores the state of a previous input and combines with the current input thereby preserving some relationship of the current input with the previous input.

Implementation is the stage of the project when the theoretical design is turned out into a working system. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve change over and evaluation of change over methods, implementation is the process of converting a new system design into operations. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system.

A. *Python programming language*

Python is a multi-programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming. Many other paradigms are supported via extensions, including design by contract and logic programming. It uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. We implement python code in spyder toolkit.

B. *Spyder toolkit*

Spyder is a powerful scientific environment written in Python and designed by and for scientists, engineers and data analysts. It features a distinct combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection and beautiful visualization capabilities of a scientific package. Furthermore, Spyder offers built-in integration with many popular scientific packages, including NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and more. Beyond its many built-in features, Spyder can be extended even further via third-party plugins. Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console or advanced editor, in your own software.

VII. CONCLUSION

Thus, the mental workload of an individual can be assessed to find the heavy workload. It can decrease the human abilities and to encounter the serious consequences. This model reaches a satisfactory classification accuracy which demonstrates workload conditions using Recurrent Neural Network. The RNN can learn spatial and spectral characteristics and layers are utilized to obtain temporal representations. In this approach, supervised training algorithm is used. When the size of training set grows, the sensitivity of the classifier typically increases while the specificity of the classifier typically decreases. It is possible that there is a lower limit of the dataset's size that will provide enough information for classification within admissible error. This model requires a large number of parameters. Future research can be validated using various data and different methods.

VIII. REFERENCES

1. P. Antonenko, F. Paas, R. Grabner, and T. van Gog, "Using electroen-cephalography to measure cognitive load," Educ. Psychol. Rev., vol. 22, no. 4, pp. 425–438, 2010, identifier: 9130.

2. Z. Wang, R. M. Hope, Z. Wang, Q. Ji, and W. D. Gray, "Cross-subject workload classification with a hierarchical bayes model," NeuroImage, vol. 59, no. 1, 2012.

3. P. Zarjam, J. Epps, and N. H. Lovell, "Beyond subjective self-rating: Eeg signal classification of cognitive workload," IEEE Trans. Auton. Ment. Dev., vol. 7, no. 4, pp. 301–310, 2015.

4. J. Zhang, Z. Yin, and R. Wang, "Recognition of mental workload levels under complex human-machine collaboration by using physiological features and adaptive support vector machines," IEEE Trans. Hum. Mach. Syst., vol. 45, no. 2, pp. 200–214, 2015.

5. R. G. Hefron, B. J. Borghetti, J. C. Christensen, and C. M. S. Kabban, "Deep long short-term memory structures model temporal dependen-cies

improving cognitive workload estimation," Pattern Recognit. Lett., vol. 94, pp. 96–104, 2017.

6. J. C. Christensen, J. R. Estepp, G. F. Wilson, and C. A. Russell, "The effects of day-to-day variability of physiological data on operator functional state classification," NeuroImage, vol. 59, no. 1, pp. 57–63, 2012.

7. L. Baldwin and B. N. Penaranda, "Adaptive training using an artificial neural network and eeg metrics for within- and cross-task workload classification," NeuroImage, vol. 59, no. 1, pp. 48–56, 2012.

8. Y. Ke, H. Qi, F. He, S. Liu, X. Zhao, P. Zhou, L. Zhang, and D. Ming, "An eeg-based mental workload estimator trained on working memory task can work well under simulated multi-attribute task," Front. Hum. Neurosci., vol. 8, 2014.

9. Phillips, M. D. Rugg, and K. J. Friston, "Systematic regularization of linear inverse solutions of the eeg source localization problem," NeuroImage, vol. 17, no. 1, pp. 287–301, 2002.

10. R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 67, no. 1, pp. 91–108, 2005.

11. Silva, J. Maltez, E. Trindade, A. Arriaga, and E. Ducla-Soares, "Evaluation of l 1 and l 2 minimum norm performances on eeg localizations," Clinical neurophysiology, vol. 115, no. 7, pp. 1657–1668, 2004.

12. Gramfort, M. Kowalski, and M. Ham¨al¨ainen,¨ "Mixed-norm estimates for the m/eeg inverse problem using accelerated gradient methods," Physics in medicine and biology, vol. 57, no. 7, p. 1937, 2012.

13. Gramfort, D. Strohmeier, J. Haueisen, M. S. Ham¨al¨ainen,¨ andKowalski, "Time-frequency mixed-norm estimates: Sparse m/eeg imaging with non-stationary source activations," NeuroImage, vol. 70, pp. 410–422, 2013.

14. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization," in Advances in neural information processing systems, 2010, pp. 1813–1821.

15. D. Strohmeier, J. Haueisen, and A. Gramfort, "Improved meg/eeg source localization with reweighted mixed-norms," in Pattern Recognition in Neuroimaging, 2014 International Workshop on. IEEE, 2014, pp. 1–4.

16. P. Xu, Y. Tian, H. Chen, and D. Yao, "Lp norm iterative sparse solution for eeg source localization," IEEE transactions on biomedical engineering, vol. 54, no. 3, pp. 400–409, 2007.

17. Z. Yin and J. Zhang, "Task-generic mental fatigue recognition based on neurophysiological signals and dynamical deep extreme learning machine," Neurocomputing, vol. 283, pp. 266 – 281, 2018.

18. Y. Ke, H. Qi, L. Zhang, S. Chen, X. Jiao, P. Zhou, X. Zhao, B. Wan, and, "Towards an effective cross-task mental workload recognition model using electroencephalography based on feature selection and support vector machine regression," Int. J. Psychophysiol., vol. 98, no. 2, pp. 157–166, 2015.

19. M. Alfaro-Ponce, A. Argelles, and I. Chairez, "Pattern recognition for electroencephalographic signals based on continuous neural networks," Neural Netw., vol. 79, pp. 88 – 96, 2016.

20. S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," Brief. Bioinform., vol. 18, no. 5, pp. 851–869, 2017.

21. Z. Jiao, X. Gao, Y. Wang, J. Li, and H. Xu, "Deep convolutional neural networks for mental load classification based on eeg data," Pattern Recognit., vol. 76, pp. 582 – 595, 2018.

22. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 25, no. 11, pp. 1998–2008, Nov 2017.

23. S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 26, no. 4, pp. 758–769, April 2018.

24. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 26, no. 2, pp. 324–333, Feb 2018.

25. J. Zhang and Y. Wu, "A new method for automatic sleep stage classifica-tion," IEEE Trans. Biomed. Circuits Syst., vol. 11, no. 5, pp. 1097–1110, Oct 2017.