***Mental Workload classifcation***

**Abstract:**

This study is attempted to classify the mental workload on EOT crane operators who operate through computers and validate various predictors and the best model for the classification. Simulated tasks of different levels were designed and used for the experiment and various indicators were measured during the experiment. The 13 indicators that were measured in the experiment are eye’s fixation frequency, fixation duration on computer screen, saccade duration, saccade amplitude, fixation to saccade ratio, MD, PD, TD, P, E, FR, Score and RT (response time). The data is obtained from SAVR lab of Industrial Engineering Department, IIT Kharapur. During the experiment two types of mental workload (Hazard and Activity) are measured and labeled as positive (high mental workload imposed) and negative (low mental workload imposed). We propose an ensemble model of a neural network based model for the classification of mental workload. Two types of mental workloads can coexist. Hence our proposed base model is a multi-label classification model.

**1.Introduction:**

The study of mental workload is one of the major part of the field of study known as ergonomics. Ergonomics is a field of science that learns the interaction and limitations of human abilities and setting up the appropriate environments and interactions with products. Ergonomics deals with the ease of working with products without harming ourselves. The main application of this field is to reduce human error, increase productivity and enhance safety and comfort by considering the interaction between the humans and the product with which they are working (or which influences the performance of humans). One such factor that is found to be influencing the productivity and performance of humans is mental workload. Mental workload can be thought of as an amount of effort put or difficulties faced to perform a task.

There is no standard definition about mental workload which is universally accepted. However Brad cain[3] defined mental workload as a mental construct, a latent variable reflecting the interaction of mental demands imposed on operators by tasks they attend to. According to Yi Ding and Yaqin Cao[2] mental workload is a subjective response to a task load that cannot be directly measured and plays an important role in human performance. Mental workload can also be defined as the mental effort put by a human to perform the given task. It could be the gap between required capabilities to perform the task and the actual effort that can be put by a healthy human. Workload can be described by the terms mental strain and emotional strain. Mental workload has a considerable influence on work performance and productivity. Hence measurement of mental workload is a crucial step in assigning tasks to enhance the best productivity in various work environments (say employees working in a company). Increased task demands and hence mental workload may lead to poor performance. Besides low levels of mental workload also leads to poor performance. Hence there is a need to maintain a certain level of mental workload and hence need to measure it. Measurement of mental workload is also a crucial part while launching a new website or product in order to get certification for use. Due to increasing technology of computers measurement of mental workload has been a crucial part in working with computers. Mental workload is a highly cognitive process and cannot be measured directly. But it can be categorized as high, low or easy, hard based on the summary of the experience given by the participant in the form of ratings. However some measures which can be measured physically called physiological processes can correlate to the mental workload. The physiological indices that have been used to measure the mental workload are electroencephalogram(EEG) activity, electrocardiographic(ECG) activity, electro-dermal activity(EDA),eye movement, respiration activities and blood pressure.

There are three types of mental workload measurement techniques proposed so far: subjective ratings or self assessment, performance measures like accuracy and productivity in the work and physiological indices which are physical responses in a human body due to the imposed mental workload. Subjective measures are taken from the operator in the form of ratings. A survey focusing on some experiences like task load,effort,mood and supporting environment to work is designed and requested to be filled by the operator for a scale of ratings(1 to 10).Even though the subjective measures are found to be useful in inferring mental workload in some cases researchers proved the disadvantages or to be precise lacking the complete information to infer mental workload. Performance measures include the level of performance of the operator in the task assigned. The performance in the task can be evaluated by various subjective measures like score, accuracy, response time, time taken to complete the task etc. The performance measures are categorized into two major types: primary task measures and secondary task measures. Primary task measures infers the operator performance on the task directly while secondary task measures infers the operator capacity to perform the task. In most of the cases secondary task measures have no practical importance compared to primary task measures. Finally as discussed earlier psychophysiological indices are the physical responses in the human body in response to the given task. Experiments had shown that physiological indices have better performances than subjective measures and performance measures in terms of sensitivity and diagnostic ability. Based on the three types of measures we can infer the information about the level of mental workload imposed on the operator. The exact mathematical relation between these measures and the level of mental workload is still unknown. However researchers interpreted this relation using various machine learning models and are satisfactorily successful in setting up a relation between them. They also interpreted the models by considering variable set of measures and by statistical analysis concluded that not all measures are of considerable importance in determining the mental workload. Also mental workload is not a unique kind of workload but it can be of various forms depending on the issue or task we are dealing with. In a particular task that we deal in this study has two types of imposed mental workload: Hazard, Activity and the measures that are considered as predictors of mental workload are eye’s fixation frequency, fixation duration on computer screen, saccade duration, saccade amplitude, fixation to saccade ratio, MD, PD, TD, P, E, FR, Score and RT (response time). The following sections includes the methodology of machine learning approach, comparative analysis of different models, various statistical plots for the data visualization, results obtained, inferences on the obtained results and finally conclusion and future scope for the study.

**2.Methodology:**

As the size of the data available is less (only 80 observations of data) a single neural network model is expected to have poor performance in predicting the mental workload. It is because of overfitting of the model to training data and weak performance on unseen data. One method to overcome this overfitting issue is the use of ensemble learning algorithms. We propose a bootstrap aggregation aka bagging model in this study. Bootstrap aggregation can be viewed as two stages: Bootstrap sampling and aggregating the results from bootstrap models. Bootstrap sampling is the process of generating a new data sample from the original data set by randomly sampling the observations from original data with replacement. This new data sample generated is called as Bootstrap sample. The size of this Bootstrap sample that is number of observations to draw from original data set is a hyper-parameter that we are free to choose. In this study we used this size same as that of the original data (ie.., 80). So our bootstrap sample may contain duplicates of the observations from the original data and some of the observations may not be included at all. Those observations that were not included in this sample are called out of box samples and will be used for testing the model as described later. Another hyper-parameter of a bagging model is the number of bootstrap samples to be generated and hence the number of classifiers used to predict the output. Now after generating the bootstrap samples a neural network model is trained on each of these bootstrap samples individually and used for new prediction. For making a new prediction we get the result from all the trained base models and aggregate them to make a final prediction. The aggregation can be made in two ways: Averaging, Max Voting. In averaging the probabilities P(y=1|X) are averaged over all the classifiers’ outputs and prediction is made based on the average(predict 1 if average probability is else predict 0 in case of binary classification). In max voting the class which is predicted by maximum number of classifiers is predicted as the output for the given feature vector. We use averaging method in our study. The summary of the process is described below.

Let M be the size of the dataset (80 here) and S be the number of subsamples or bootstrap samples to be drawn from the original dataset. N(=13) be the number of predictors or features used for training. The original dataset is { ( , ),( , ),( , ),…..( , ) }.

**1)**Preprocessing the data: Feature scaling is needed to be done on each of the predictors in feature vector X. Z-score normalization or Standardization is used for this purpose.

for each ∈ where =1,2,3,…..,N

Output vectors were also modified to represent in binary form. In the given data, classes are represented as 1,2. All classes that belong to 2 were converted to 0’s. For example, an entry 0,1 in Hazard column means Hazard belongs to classes 2,1 respectively (similarly for Activity column).

**2)**Bootstrapping and training the model: Generate a bootstrap sample of size 80 { (, ), ( , ),…( ,) } where i=1,2,…,S from original data with replacement. Build a neural network model with one hidden layer having 10 units and an output layer with two units for two classes. Sigmoid activation function is used instead of softmax activation function at the output layer because it is a multi-label classification. Train the model on the generated bootstrap sample and store the model in a list of base models(M). Find the out of box samples from the original dataset that were not used for training this model. Maintain a list (named OOB) of 80 lists (for each observation) and for each of these out of box observations add the classifier’s index into its corresponding list in the list L. Repeat this procedure for S number of bootstrap samples.

The list of models M=[] and let hypothesis function of each corresponding model is

**3)** Prediction and evaluating the model: For each observation in the data evaluate the output probability for each classifier that did not include this observation while training (using list OOB) and take average of them.

P(Y=[1,1]|X(i))=

In general equation the above equation is used for predicting the output for unseen data in test set where the summation runs over all the classifiers. The study by Breiman on error estimates of bagged classifiers proved that out of bag estimate is as accurate as using test set estimates in evaluating the performance of the model.

Data set

Subsample S

Subsample4

Subsample3

Subsample2

Subsample1

Trained neural network S

Trained neural network 4

Trained neural network 3

Trained neural network 2

Trained neural network 1

Integration-

Meta classifier

Output

Figure 1: Flow chart of ensemble learning (bootstrap aggregation) algorithm

Pseudocode for ensemble technique with single layered neural network using 10 bootstrap samples:

1.Read and preprocess the data🡪feature scaling

2.divide the data set into input data (X) and output (y)

3.classifiers🡨10,OOB\_list🡨[[],[],……80 empty lists], M\_list🡨[],tloss🡨0,tacc🡨0

4.for i🡨1 to classifiers

Build a neural network of an input layer (13 features), hidden layer (10 units), output layer (2 units)

Create empty data sets Xtrain and ytrain having same shape as X and y respectively

boot\_indices🡨list of random numbers from 1 to 80

for j in boot\_indices

draw the sample from X and y having index=boot\_indices[j]

Xtrain[j],ytrain[j]🡨sample drawn in above step

create list oob\_indices=indices from 1 to 80 that are not present in boot\_indices

for index in oob\_indices

OOB\_list[oob\_indices[index]].append(i)

fit the model to (Xtrain,ytrain) and add it to M\_list

tloss+=training loss of the model

tacc+=training accuracy of the model

tloss/=classifiers,tacc/=classifiers

predictions🡨empty numpy array of shape of y

for i🡨1 to 80:

yhat=[0,0]

for j in oob\_list[i]:

pred=M\_list[oob\_list[i][j]].predict(X[i,:])

yhat+=pred

yhat=yhat/length(oob\_list[i]) //average prediction

yhat=(yhat>=0.5)\*1

predictions[i,:]=yhat

accuracy=accuracy\_score(predictions,y)

hammingloss=hamming\_loss(predictions,y)

confusionmatrix=multilabel\_confusion\_matrix(predictions,y)

precision=precision\_score(predictions,y,average='macro')

recall=recall\_score(predictions,y,average='macro'))

f1score=f1\_score(predictions,y,average='macro'))

From the average result predict the classes for mental workload. These predictions are compared to the actual labels and the model is evaluated with various evaluation metrics: accuracy, hamming loss, confusion matrix, Precision, recall and F1 score.

Accuracy=

Hamming loss=

where is the actual value(class) and is the predicted value(class) and L=size of output=2

Confusion matrix= where TN=True negative TP=True positive,

FN=False negative FP=False positive

Precision=

Recall=

F1 score=

Individual confusion matrices are obtained for different labels=[Hazard, Activity] and there individual precision and recall are calculated. For evaluating precision, recall and f1 score for multi-label classification individual precisions and recalls are calculated for the two classes from their confusion matrices. There are two ways to report precision, recall and f1 score: Micro averaging and Macro averaging. We use Macro averaging for this study.

Macro average precision=P=

Macro average recall=R=

Macro average F1 score=

**3.Data Visualization:**

Data visualization is the process of generating various plots like box plot, scatter plot and many other statistical plots to have a better insight about the distribution of the available data set. These processes help us to decide about the model and structure that gives the best performance in predicting the classes without actually going for cross validation set. Box plot is a way to display the distribution of data based on its limits and quartiles (minimum, first quartile(Q1), median, second quartile(Q2), maximum) and it gives the insight about the variability about a particular feature in predicting a class. It can also give us an insight about the symmetry, skewness and how tightly the data is distributed.

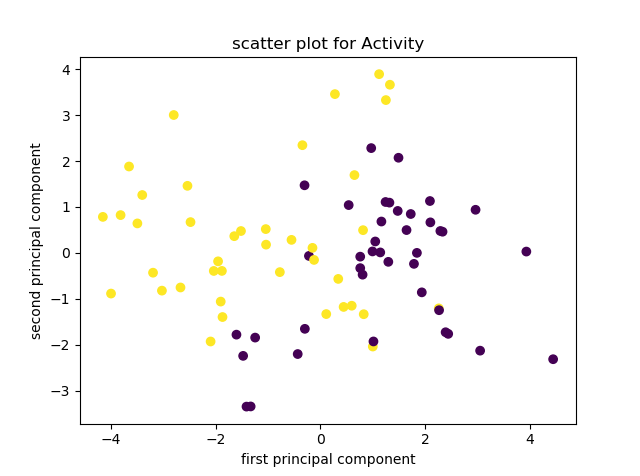
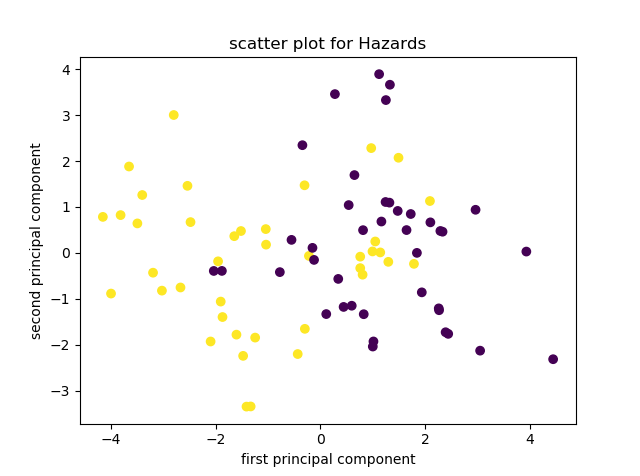
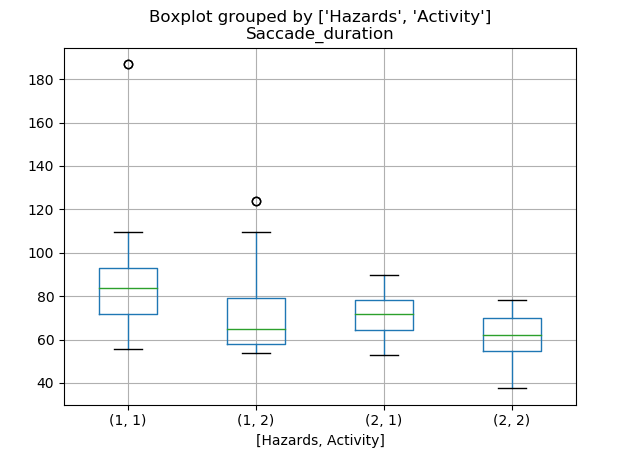
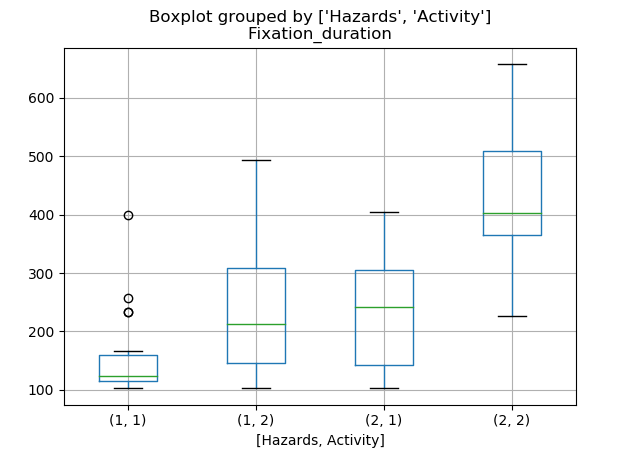
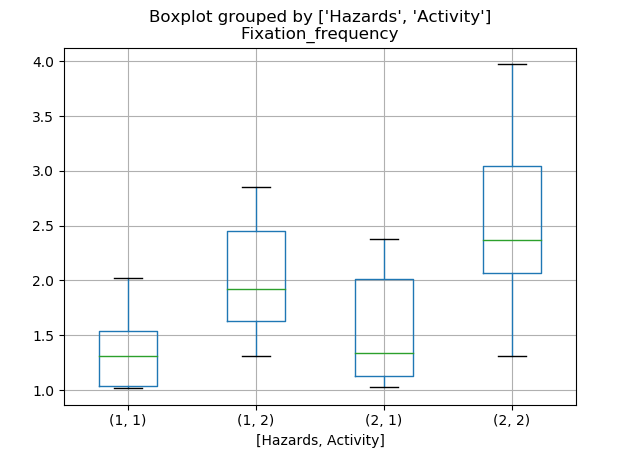
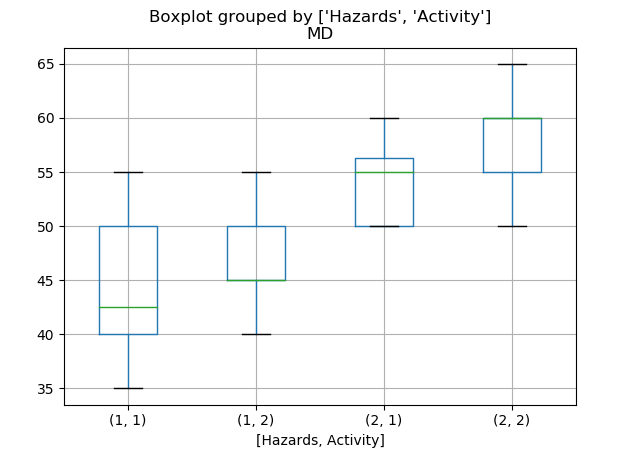
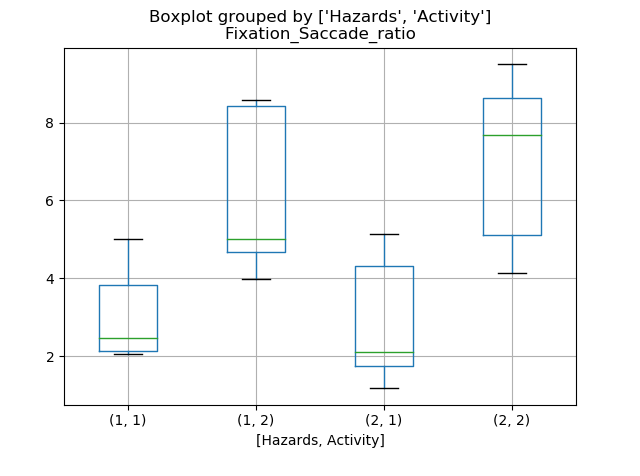
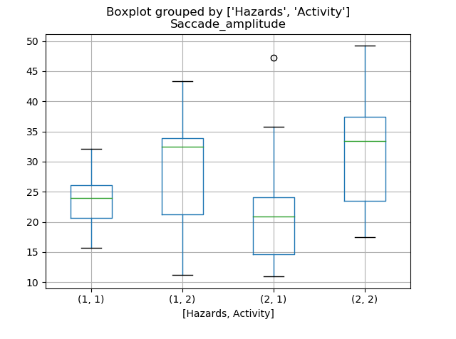
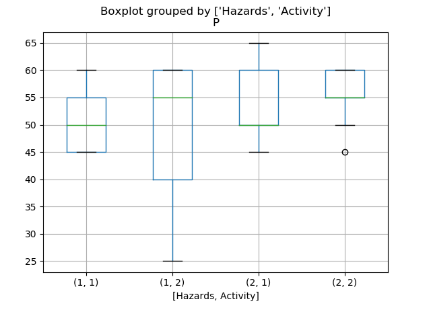
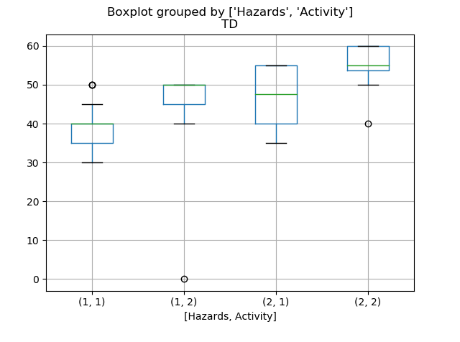
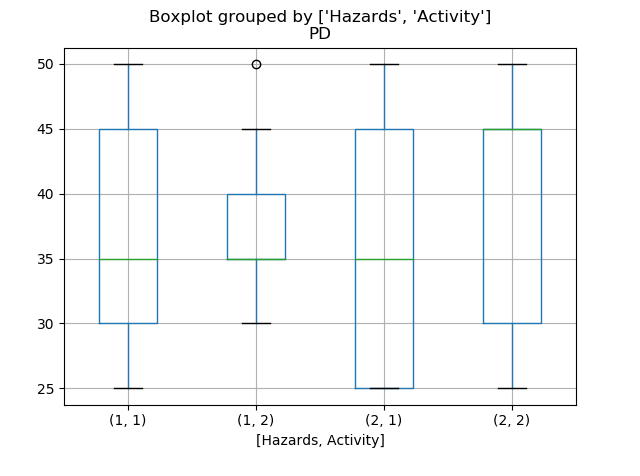


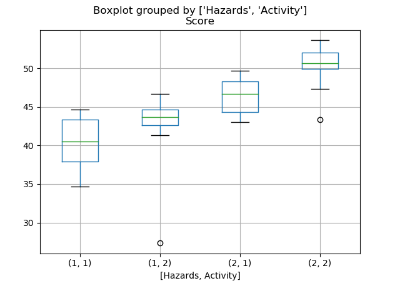
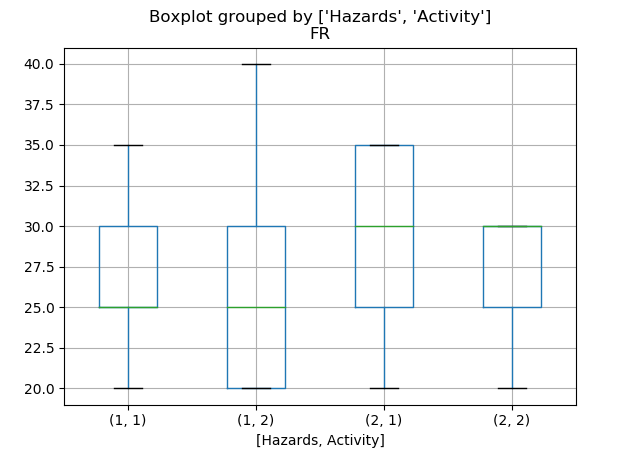
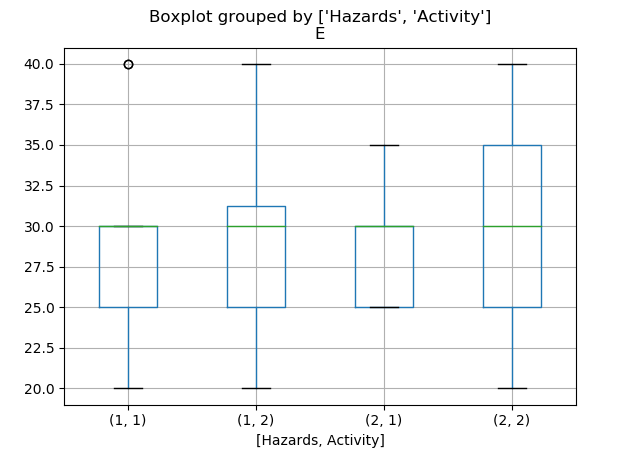
Figure 2: Scatter plots between two principal components for(a)Hazard, (b)Activity purple dots indicate class=1 and yellow dots indicates class=2(for both hazard and activity)

Scatter plots are used to plot data points on a horizontal and a vertical axis to show to what extent and how a variable is related to another variable and visualize the decision boundary. The decision boundary is a curve or boundary which differentiates between two classes on two dimensional scatter plot. The shape of the decision boundary may help in deciding the appropriate model to use to classify the classes. Scatter plots show the correlational structure between the variables. The scatter plots of the data set are shown in figure 2. The 13 dimensional feature vector is reduced to 2 dimensional vector by using principal component analysis(pca) algorithm to make it possible to plot in two dimensional space. It can be seen from the scatter plot that the decision boundary is not linear and hence the correlation between the features non-linear. This give rise to the fact that neural network is a good classifier than a simple logistic regression whose decision boundary is linear. The box plots are shown in figure 3. We can observe that there is high variability in PD for Hazard=1 and Activity=1 and least variability is observed in Score for Hazard=1 and Activity=2.









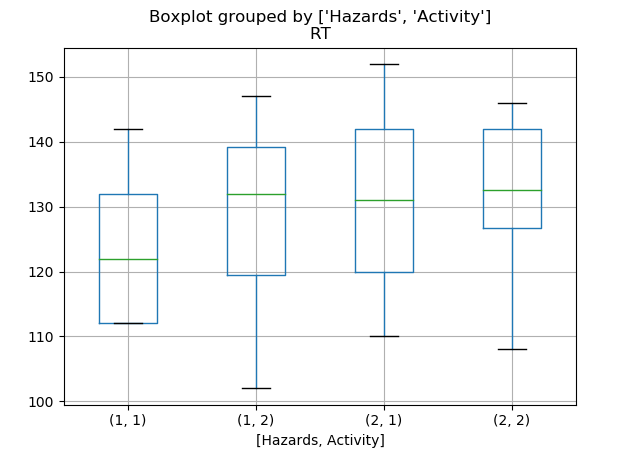


Figure 3: Box plots for different features for four types results (two classes)

**4.Results and comparisions:**

4.1:Neural network based ensemble technique for three different models:

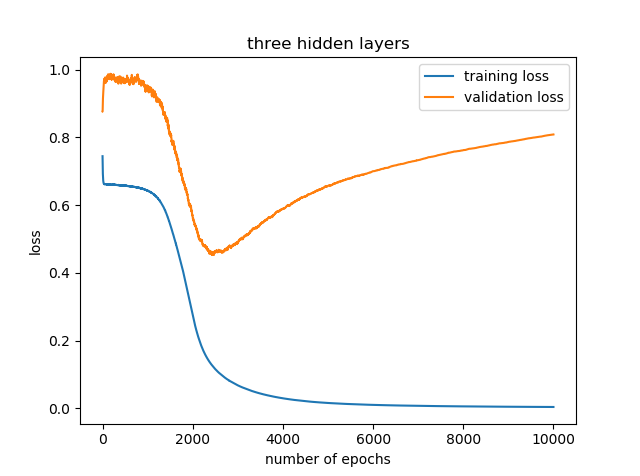
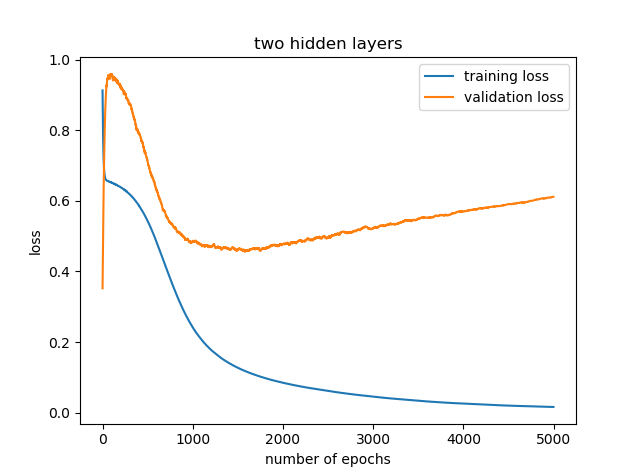
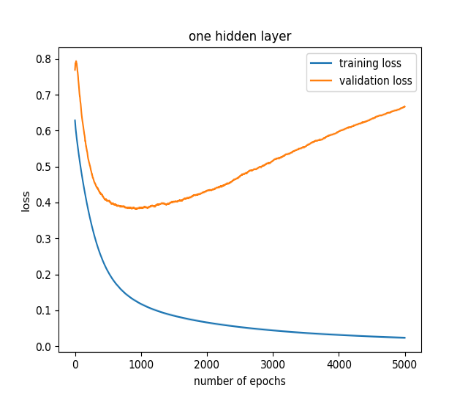
A single neural network model without using bootstrap technique was trained on 80% of the data and using 20% of the data as cross validation set learning curves (loss vs number of epochs) were plotted. The plots are shown in figure 4. The plots suggest that maximum number of epochs to train the model to avoid overfitting are 1000,1500,2500 for one,two,three hidden layered neural networks respectively. 

Figure 4:learning curves for three different models

Three different bagging models with a neural network having one, two and three hidden layers as a base model were trained on the given data. In each of the neural network, input layer contains 13 features, each hidden layer contains 10 units and the output layer contains two units. Then the entire data set is again used to test the model. For each observation in the data set the prediction is made from only those classifiers for which the observation is out of box sample (it is not used for training the particular classifier). The final prediction is made using the averaging method of aggregation. The prediction set and original set of classes are then compared to evaluate the model using the evaluation metrics: accuracy, hamming loss, confusion matrices, Precision, Recall, F1 score. The results are shown in tables 1,2,3. Variation of the evaluation metrics with number of classifiers are shown in figure 5 and with number of hidden layers are shown in figure 6. The maximum accuracy of 86.25% is obtained for single layered neural network with 40 classifiers where they are 82.5 and 85 for two and three layered networks respectively. The hamming loss (0.08125) is minimum for single layered neural network whose values are 0.09375, 0.875 for two and three layered networks respectively. The maximum precision of 0.9375 is obtained in single and two layered neural network where it is 0.925 for three layered network. The maximum recall of 0.9039 is obtained again with 40 classifiers in single layered neural network whose values for two and three layered neural networks are 0.9, 0.884 respectively. At last the maximum f1 score of 0.92 is obtained again for single layered neural network with 40 classifiers where they are 0.90 and 0.9155 for two and three layered neural networks respectively. Hence it can be concluded that the neural network based model with one hidden layer has the best performance among the three models. This could be happening because of overfitting or high variance in those models. Also adding more layers in neural network increase the complexity and computational cost of the model. Hence the best model for classification of mental workload is a single layered neural network based model. Another conclusion that can be made is that among two and three layered neural network models two layered network has the better performance than the other one. The same reason of increase in variance with increase in hidden layers than required may be leading to this difference among two and three layered networks.

**►*Table 1: Neural network with one hidden layer***

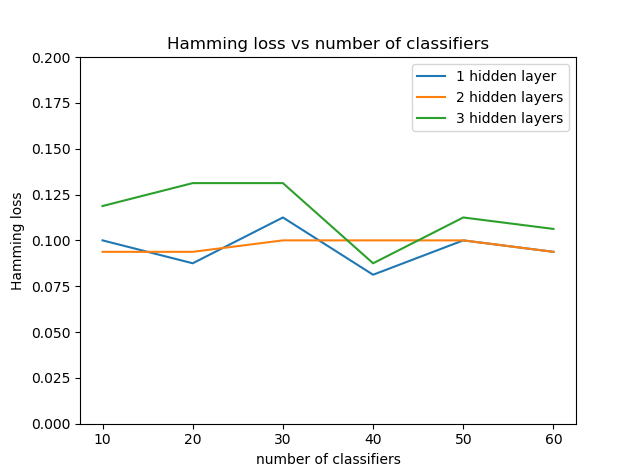
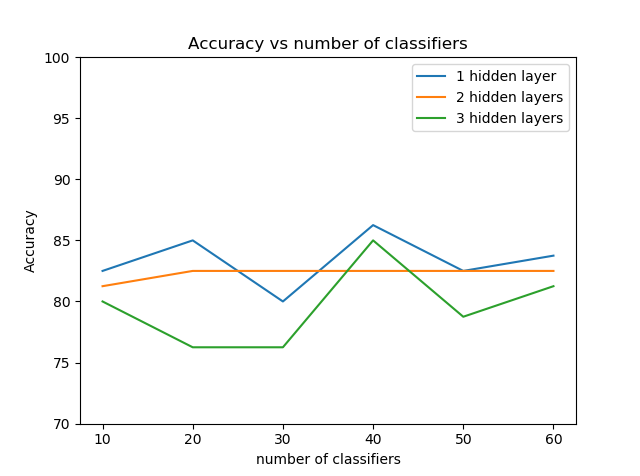
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number of classifiers | accuracy | Hamming loss | Confusion matrix for label=[Hazard] | Confusion matrix for label =[Activity] | Precision  (Macro average) | Recall  (Macro Average) | F1 score  (Macro average) |
| 10 | 82.5 | 0.1 | () | () | 0.925 | 0.88 | 0.9024 |
| 20 | 85 | 0.0875 | () | () | 0.9375 | 0.8928 | 0.9146 |
| 30 | 80 | 0.1125 | () | () | 0.9125 | 0.8687 | 0.8899 |
| 40 | 86.25 | 0.08125 | () | () | 0.9375 | 0.9039 | 0.92 |
| 50 | 82.5 | 0.1 | () | () | 0.925 | 0.88 | 0.9024 |
| 60 | 83.75 | 0.09375 | () | () | 0.9375 | 0.8823 | 0.909 |

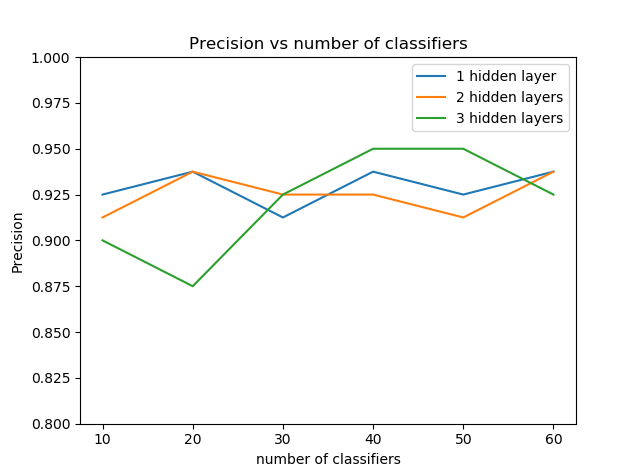
**►*Table 2: Neural network with two hidden layers***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number of classifiers | accuracy | Hamming loss | Confusion matrix for label=[Hazard] | Confusion matrix for label =[Activity] | Precision  (Macro average) | Recall  (Macro Average) | F1 score  (Macro average) |
| 10 | 81.25 | 0.09375 | () | () | 0.9125 | 0.9 | 0.907 |
| 20 | 82.5 | 0.09375 | () | () | 0.9375 | 0.883 | 0.909 |
| 30 | 82.5 | 0.1 | () | () | 0.925 | 0.881 | 0.902 |
| 40 | 82.5 | 0.1 | () | () | 0.925 | 0.881 | 0.902 |
| 50 | 82.5 | 0.1 | () | () | 0.9125 | 0.89 | 0.901 |
| 60 | 82.5 | 0.09375 | () | () | 0.9375 | 0.8823 | 0.909 |

**►*Table 3: Neural network with three hidden layers***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number of classifiers | accuracy | Hamming loss | Confusion matrix for label=[Hazard] | Confusion matrix for label =[Activity] | Precision  (Macro average) | Recall  (Macro Average) | F1 score  (Macro average) |
| 10 | 80 | 0.11875 | () | () | 0.9 | 0.869 | 0.884 |
| 20 | 76.25 | 0.13125 | () | () | 0.875 | 0.864 | 0.868 |
| 30 | 76.25 | 0.13125 | () | () | 0.925 | 0.831 | 0.8756 |
| 40 | 85 | 0.0875 | () | () | 0.95 | 0.884 | 0.9155 |
| 50 | 78.75 | 0.1125 | () | () | 0.95 | 0.845 | 0.894 |
| 60 | 81.25 | 0.10625 | () | () | 0.925 | 0.87 | 0.897 |





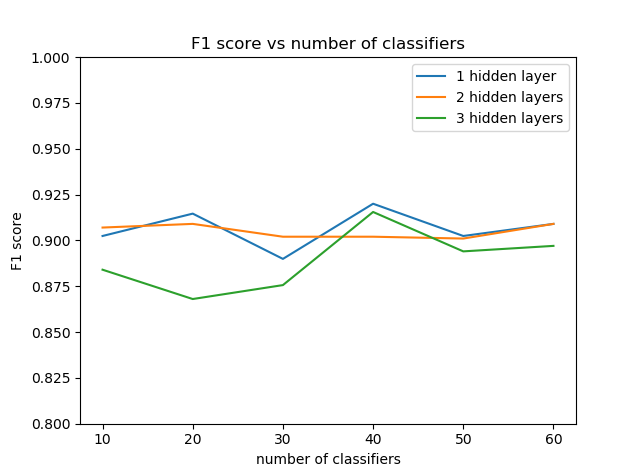
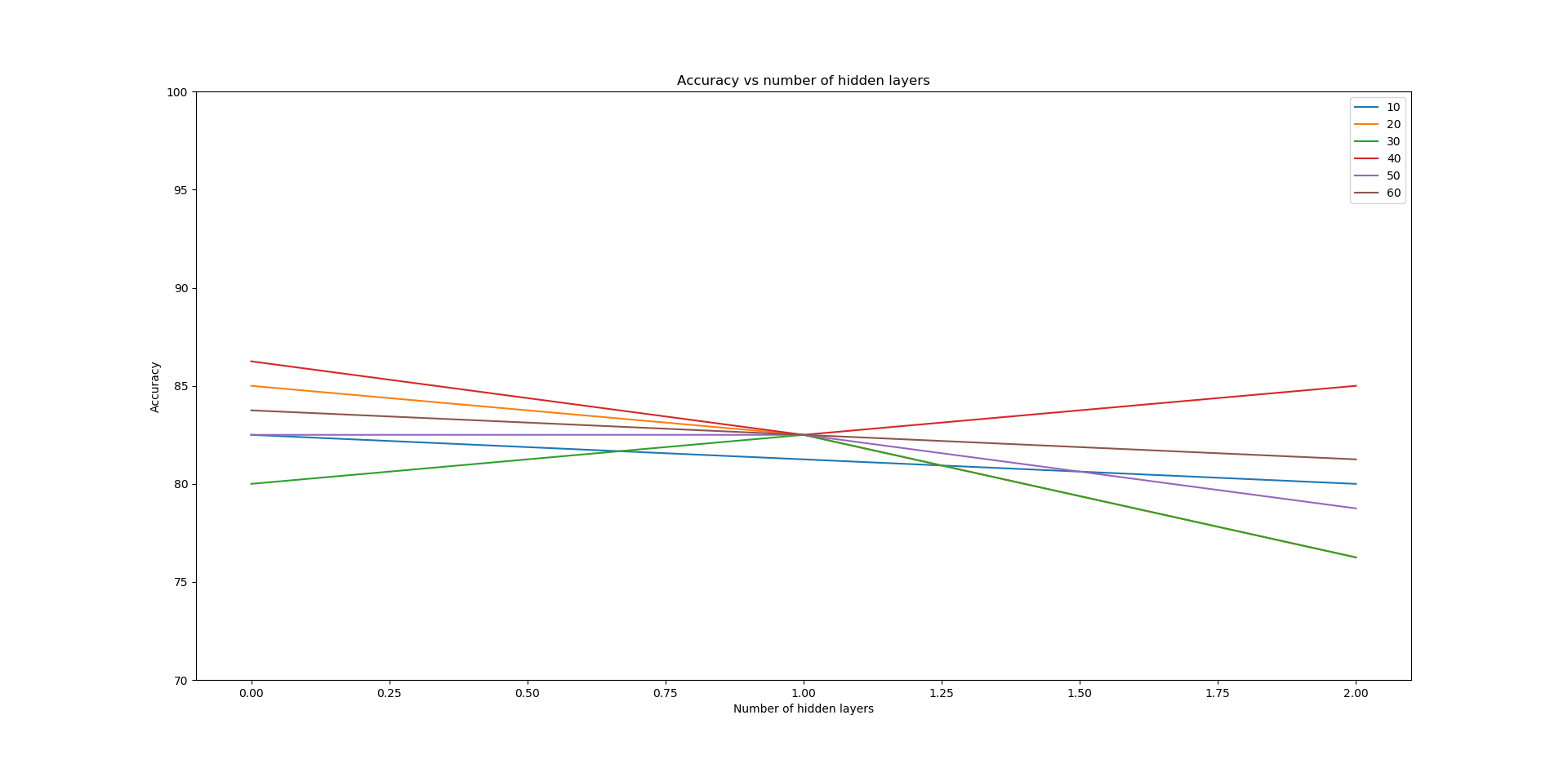
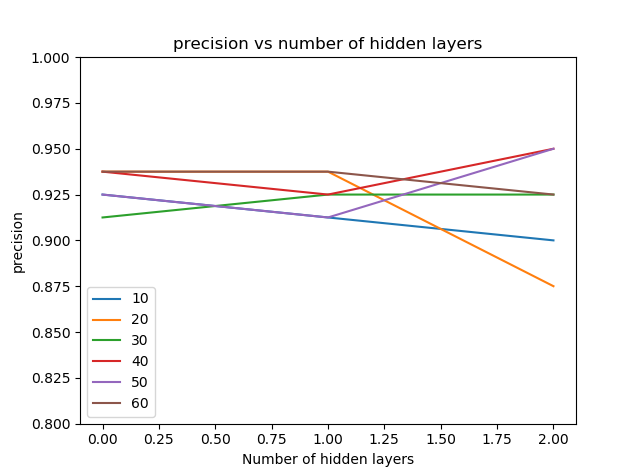
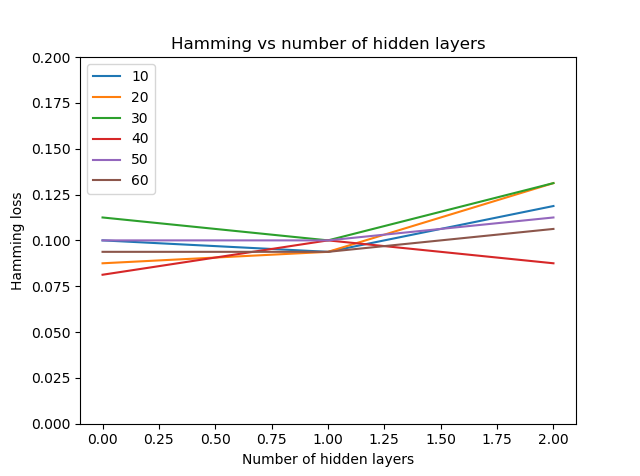


Figure 5:(a) Accuracy (b)Hamming loss (c)Precision (d)Recall (e)F1 score vs number of classifiers with three models M1, M2, M3





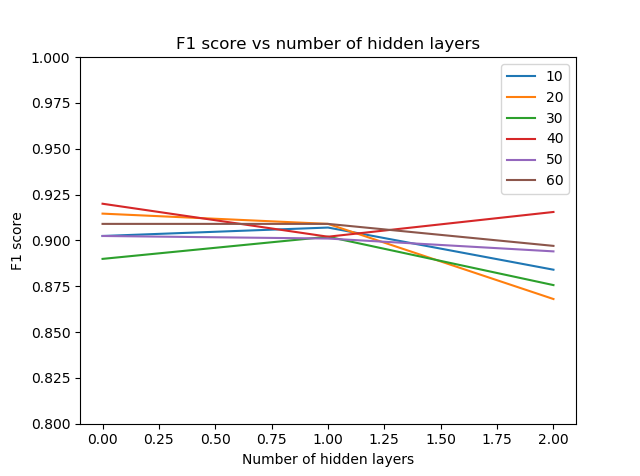
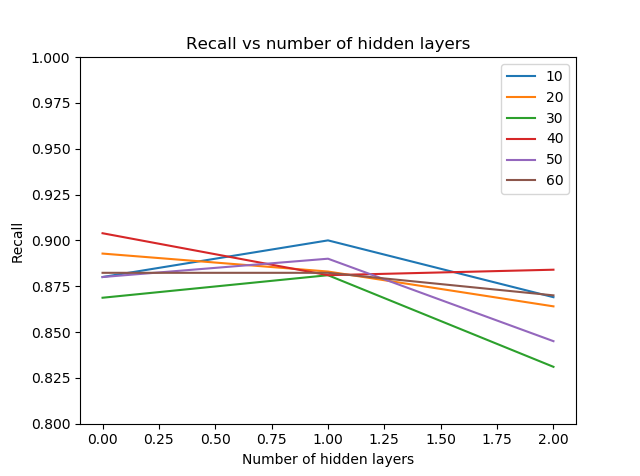


Figure 6:(a) Accuracy (b)Hamming loss (c)Precision (d)Recall (e)F1 score vs number of hidden layers for different number of classifiers

4.2: Ensemble technique with logistic regression:

Eventhough we propose a neural network model to classify the mental workload using the given data there is a need to look up for alternative models and compare if it has better performance than our proposed model. We build a model with the same ensemble technique as our proposed model but with logistic regression as the base model instead of a neural network. The performance of the logistic regression based model with the same process of bootstrap sampling, training a model on it and testing it on the out of box samples is evaluated and shown in table 4. The maximum accuracy of 82.5% is obtained when using 50 classifiers whereas for neural network models with 1,2,3 hidden layers the maximum accuracy of 86.25, 82.5, 85 are obtained respectively. The average accuracy over the number of classifiers for logistic regression based model, neural network model with 1,2,3 hidden layers are 79.58, 83.33, 82.29, 79.58 respectively. These values draw a conclusion that neural network based models have better performance than logistic regression based model. This is in accordance with our conclusion from the nature of decision boundary in the scatter plots of the data in data visualization section. Similar trend can also be observed in the variations of hamming loss, precision, recall and f1score. The average estimates of hamming loss are 0.1135, 0.0958, 0.0968, 0.1145 respectively and f1 scores are 0.8897, 0.9064, 0.905, 0.889 respectively. This shows that neural network based models perform better than a simple logistic regression based model. Moreover the performance is decreasing among the neural network models with increase in number of hidden layers.

►**Table 4: Logistic regression based ensemble technique**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number of classifiers | accuracy | Hamming loss | Confusion matrix for label=[Hazard] | Confusion matrix for label =[Activity] | Precision  (Macro average) | Recall  (Macro Average) | F1 score  (Macro average) |
| 10 | 73.75 | 0.14375 | () | () | 0.875 | 0.8427 | 0.8582 |
| 20 | 80 | 0.1125 | () | () | 0.9125 | 0.8687 | 0.8899 |
| 30 | 80 | 0.1125 | () | () | 0.9375 | 0.8524 | 0.8928 |
| 40 | 81.25 | 0.10625 | () | () | 0.925 | 0.8708 | 0.8968 |
| 50 | 82.5 | 0.09375 | () | () | 0.9375 | 0.8826 | 0.909 |
| 60 | 80 | 0.1125 | () | () | 0.925 | 0.8604 | 0.8914 |

**4.Discussion:**

The present study investigates measurement and identification of mental workload during computer simulated tasks. Previous studies focused on the classification of the single type of mental workload into their levels like easy, medium and hard. But the mental workload is not unique type. It can be imposed in many different ways and hence the definition of mental workload can be in many ways. These workloads can exist simultaneously on the operator. Hence there is a need to study and classify the mental workloads that are simultaneously imposed on the participant. Our study is mainly focused on this type of classification of mental workloads. The data is obtained from the SAVR lab and used for our study. We implemented various models to fit for the data and finally proposed an ensemble model with base model being a neural network having a single layer as a best fit for the classification of mental workload. We elaborated the performance of neural network models with 1,2,3 hidden layers, logistic regression model and Bayesian neural network model to draw this conclusion. A single base model gave an accuracy of around 75% in average where using bootstrap aggregation on these individual models on the same data gave an accuracy greater than 80%. Precision, recall and f1 score were also high in bootstrap aggregation than a single base model and also the hamming loss is low in former model. Hence there is an increasing effect on the performance of the model by using bootstrap aggregation which itself is the main agenda of the model. For single hidden layered network the maximum performance is obtained when using 40 classifiers/bootstrap samples. It can be seen from the evidence in the peak in accuracy, precision, recall, F1 score and the least hamming loss at 40 classifiers in the plots of these metrics vs number of classifiers. The accuracy, hamming loss, precision, recall and f1 score are 86.28%, 0.08125, 0.9375, 0.9039, 0.92. The same measures for two layered network are 82.5%, 0.09375, 0.9375, 0.883, 0.909 and for the three layered network are 85%, 0.0875, 0.95, 0.884, 0.9155. The accuracy is highest and the hamming loss is least in single layered neural network. The precision is highest in three layered neural network. The recall and f1 score are highest in single layered neural network. These results suggest that single layered neural network has the best performance of the three models. Results on alternate models like logistic regression also showed the poor performance in comparision to our proposed model. The Bayesian neural network also has many disadvantages and it may fail to accurately classify the mental workload.

A Bayesian neural network is comprised of a probabilistic model and a neural network. The probabilistic model incorporate random variables and probability distributions into the model of an event. It gives a probability distribution as an outcome. Random variables from Normal, Binomial and Bernoulli distributions form the basis of these models. Besides predicting the class for input feature vector it also shows the confidence level of the model in prediction where in simple neural network model it only predicts the output class. A simple neural network is a deterministic model where Bayesian neural network is a probabilistic model. It gives a single possible outcome which is the predicted class as output. A Bayesian neural has advantages over a simple neural network but in most cases it fails in prediction. There is no proper theoretical justification for Bayesian neural network from a probabilistic perspective. While training a Bayesian neural network likelihood estimation is calculated and the model converges towards maximization of likelihood where in simple neural network loss function (binary cross entropy) is calculated and the model converges towards the minimization of loss function. Using maximum likelihood estimation may ignore any uncertainty in the weights of the model. This type of training is often susceptible to overfitting. The correct thing to do is the estimation of posterior inference but they have to be trained by approximations such as Markov Chain Monte Carlo(MCMC) methods. The Bayesian neural network may not converge to global maximum of likelihood function using MCMC method for training. This is because the path of convergence not along the increasing direction of likelihood function. The training algorithm selects a point in the vicinity of present weights according to some distribution with mean located at present point and having some variance. It then evaluates the likelihood of the input vector at both the points and then selects whether or not to move to the new point according to the following criteria.

If , are the initial and final weights of a step in MCMC method then if

Where U (0,1) is a random value between 0 and 1 sampled according to uniform distribution.

The training of Bayesian model using MCMC method needs many number of iterations to converge than in simple neural network’s gradient descent method. The MCMC method moves around the global maximum but may not be present at the global maximum after reaching it where in gradient descent once it reaches to global minimum it remains at that position as the gradient term becomes zero and the updating step has no effect on the weights. The gradient descent algorithm is described below.

Gradient descent () {

Repeat {

Compute J()

Compute

//simultaneous update

} }

where is the matrix of weights mapping layer to layer +1 and

J()= binary cross entropy =

and is the learning rate of the gradient descent algorithm.

In Bayesian neural networks training features are assumed to be independent. This is not acceptable in many cases and hence the performance in unsatisfactory. So we chose simple neural network over Bayesian neural network for the classification of mental workload in this study. Hence our proposed model which is a single hidden layered neural network based ensemble model is the best model for the classification of mental workload.

Our results also showed that hamming loss is a better evaluation metric than the most used accuracy metric. The accuracy measure as evaluation metric impose strict expectations on the model than using hamming loss as evaluation metric. This is because accuracy metric considers fraction of observations that are predicted correctly in both the classes. It doesn’t consider partial predictions (that is one class predicted correctly but another class predicted wrong) into consideration. But hamming loss considers these partial predictions into consideration. It can be seen that accuracy is around 80 to 85% for all the models. Whereas hamming loss is around 0.1 which means 90% of the classes were predicted correctly. Hence using hamming loss as evaluation metric over accuracy gives us the better understanding about the model.

Also it should be noted that precision, recall and hence F1 score were calculated by macro averaging the respective metrics of individual classes. Another way of averaging is micro averaging. It is expected that the results would be similar to macro averaging method. The performance of the two layered neural network is almost stable with number of classifiers. That means there is not much effect on performance by using bootstrap aggregation after 10 bootstrap samples. For three hidden layers the trend is similar to one hidden layered network. The peak performance is observed at 40 classifiers. Besides the performance of three hidden layered network is lower when compared to one hidden layered network. This can also be observed in the plots of the evaluation metrics with number of hidden layers in Figure 5.

**5.Conclusion:**

The study researched about the two types of mental workload, Hazard and Activity on a computer operator. It investigated the performance of various possible models and finally concluded that neural network with one hidden layer having 10 units as the best model for the classification of mental workload. It also removed the possibility of using logistic regression for classification by studying decision boundary in scatter plots and also results from its implementation. It also justified the agenda of bootstrap ensemble technique of increasing the performance of the overall model by reducing the variance in the model. It explained why probabilistic models like Bayesian neural network is not used for the classification of mental workload. Finally, the study is successful in achieving about 85% accuracy with a very small data set. Further studies can be made by obtaining more data and may achieve higher accuracy and performance by training the model using the data. The methods used in this study and the findings may be applied to various products other than computers and can optimize their design.

**6.Contributions, Limitations and future work:**

Although previous studies on mental workload are successful in identifying and studying the impact of workload they did not consider the chance of co-existence of different types of mental workloads. This paper contributes to the classification of mental workload into two types of coexisting mental workloads. The contributions of this study can be applied in many industrial areas in optimizing their products. They can also be used on employees in a company to attain the best performance by optimizing the mental workload. The limitations of the study are the lack of accurate technology to obtain more amount of data or proper predictors considered to classify the mental workload. Future work can be made on applying various different models to check for higher performance in the classification of mental workload. Some of the models that can be tried would be using Bayesian network, SVM, deep learning networks like RNN, CNN, LSTM etc.

**References:**

[1] Jessica Wilson (2019). What is mental workload [And how to measure it]. https://imotions.com/blog.

[2] Yi Ding, Yaqin sao, Vincent G Duffy, Yi Wang &Zhang Xuefeng (2020). Measurement and identification of mental workload during simulated computer tasks with multimodal methods and machine learning.

[3] Cain, Brad. (2007). A Review of the Mental Workload Literature. English. 35.

[4] Karthik Nooney (2018). Deep dive into multi-label classification (With detailed case study). Blog-towards data science. https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff

[5] Jason Browniee (April 22,2016). Bagging and Random Forest Ensemble Algorithms for Machine Learning. Blog-Machine learning mastery. https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning

[6] V.K.Rohatgi. An introduction to probability and statistics. Second edition

[7] Christopher M.Bishop(2006). Pattern recognition and machine learning. First edition

[8] Vikram Mullachery, Anirudh Khera and Amir Hussain. A study of Bayesian neural networks. https://arxiv.org/ftp/arxiv/papers/1801/1801.07710.pdf

[9] Joseph Rocca (2019). Ensemble methods: bagging, boosting and stacking. Blog-towards data science. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

[10] Jason Browniee(November 11,2016). How to Implement Bagging From Scratch With Python. Blog-machine learning mastery. https://machinelearningmastery.com/implement-bagging-scratch-python

[11] Leo Breiman. Statistics department, University of California. Out of Bag Estimation. https://www.stat.berkeley.edu/~breiman/OOBestimation.pdf

[12] Tavish Srivastava(2015). Basics of Ensemble Learning Explained in Simple English. Blog-analytics vidhya. https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning