Summary of the activities and the result:

In this exercise, a simple Artificial Neural Network classifier were used to split the data set into two classes, i.e., "stress"="1" and "rest"="2". First, the data was loaded into MATLAB and the data was split into two parts 1000 first samples were used to train the classifier, validate and test. And 420 remaining samples were used and unseen data to test ultimately the classifier performance. Off course, the classifier selection, training, validation and testing can be done in many ways some with very precise results and time consuming, others with low precision but fast. Depending on the spec and the requirements of the application the classifier can be selected, however in this classification practice none of the above were given as input. Therefore one classifier, which is ANN, was selected randomly.

The architecture of the Perceptron classifier is as shown in Fig.1.
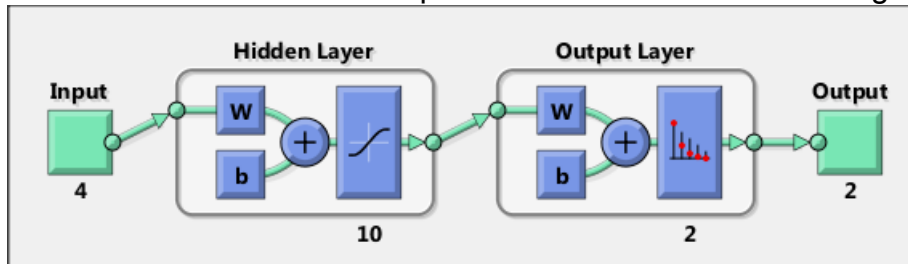

Fig.1 ANN classifier architecture

The classifier was trained using 1000 samples (from the beginning of the data block). 75% for training, 15% cross validation and 15% for testing. The gradient decent and back propagation training techniques were used to train the classifier. The training error of the classifier versus number of training is shown in Fig.2. It is clearly shown that the curve is gradient decent.
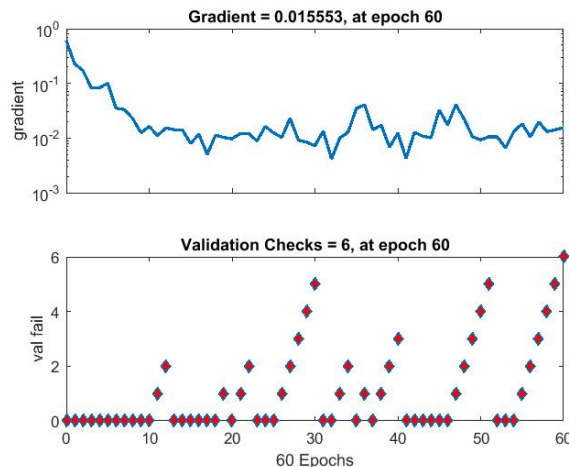

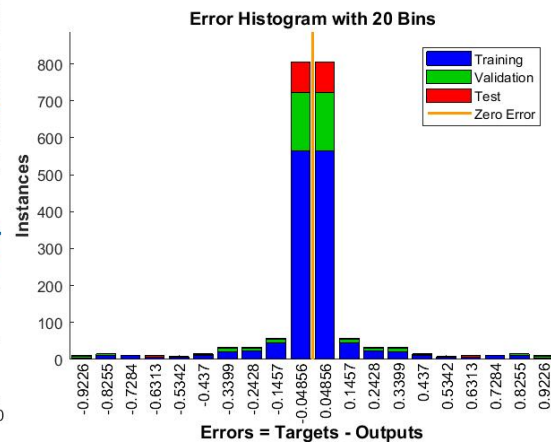Fig.2 gradient of training

Fig.3 error histogram

The error histogram of the training is show in Fig.3

After this step, 15% of the data used to validate the classifier precision. This precision is defined by the ROC curve and confusion matrix analysis. These analyses is shown in Fig.4.

Fig.4 confusion matrix for validation error analysis

At the end, after training and test of the classifier 30% remaining data (420 samples) were presented to the classifier and noticed good ROC curve and very low classification error. These results are shown in Fig.5. The MATLAB code and data and the classifier training weight can be find in the GitHub repository. https://github.com/SepidehHadadi/classificaiton_stage_example

your can find more data about my previous work in the same GitHub link: https://github.com/SepidehHadadi

Confusion Matrix

	Target Class 1	Target Class 2	
Output Class 1	191 / 45.5%	7 / 1.7%	96.5% / 3.5%
Output Class 2	8 / 1.9%	214 / 51.0%	96.4% / 3.6%
	96.0% / 4.0%	96.8% / 3.2%	96.4% / 3.6%

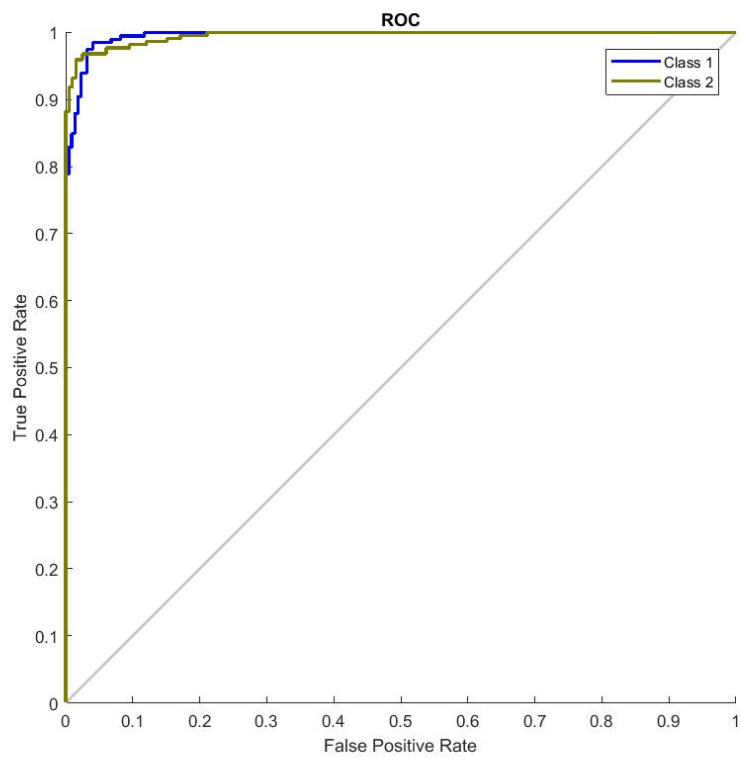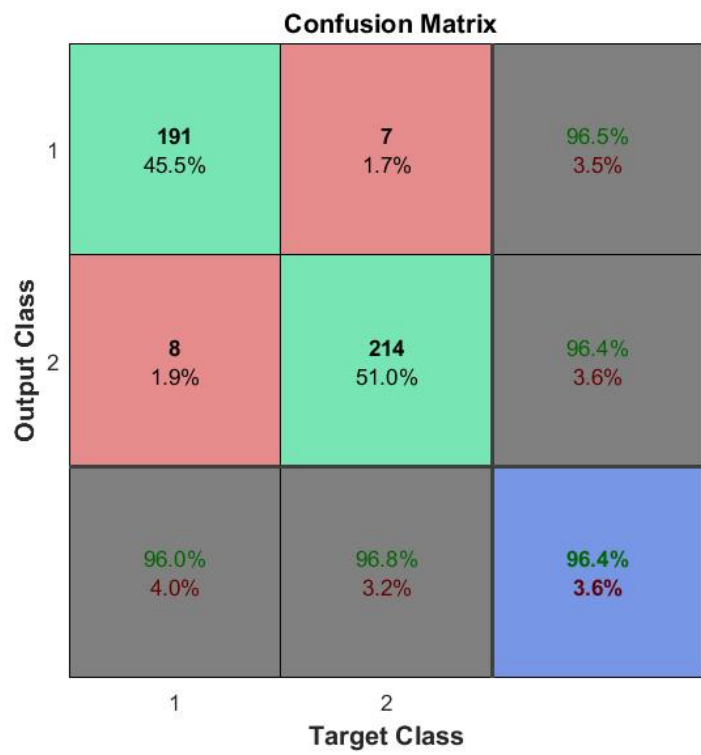## ROC

True Positive Rate vs False Positive Rate

Legend: Class 1, Class 2

Fig. Final classification result for unknown data