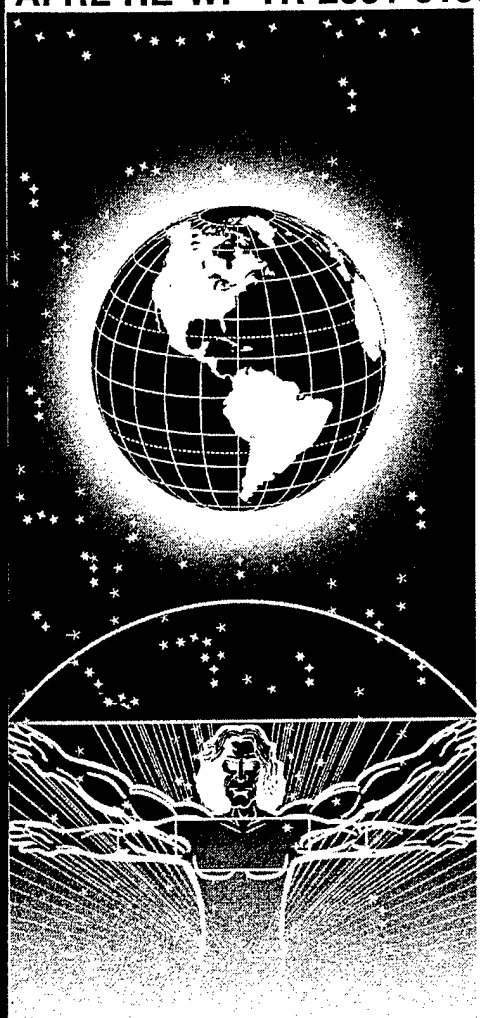
AFRL-HE-WP-TR-2001-0136



UNITED STATES AIR FORCE RESEARCH LABORATORY

SELECTING SALIENT FEATURES OF PSYCHOPHYSIOLOGICAL MEASURES

Chris A. Russell

HUMAN EFFECTIVENESS DIRECTORATE
CREW SYSTEM INTERFACE DIVISION
WRIGHT-PATTERSON AFB, OHIO 45433-7022

Steven G. Gustafson

SCHOOL OF ENGINEERING AND MANAGEMENT
DEPT OF ELECTRICAL & COMPUTER ENGINEERING
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AFB, OHIO 45433

JUNE 2001

INTERIM REPORT FOR THE PERIOD JANUARY 2001 TO JUNE 2001

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB OH 45433-7022

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, Virginia 22060-6218

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2001-0136

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

MARIS M. VIKMANIS
Chief, Crew System Interface Division
Human Effectiveness Directorate
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2001	Interim Report, January 2001 to June 2001

4. TITLE AND SUBTITLE

Selecting Salient Features of Psychophysiological Measures

5. FUNDING NUMBERS

PE 62202F
PR 7184
TA 08
WU 64

6. AUTHOR(S)
*Chris A. Russell
** Steven G. Gustafson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7022

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-HE-WP-TR-2001-0136

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

*Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7022

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

** School of Engineering and Management, Dept of Electrical and Computer Engineering
Air Force Institute of Technology, Wright-Patterson AFB, OH 45433

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

Determining operator cognitive or functional state is a critical component of adaptive aiding systems. To determine cognitive state, we must decide which measured features from the human will assist in distinguishing different levels of mental activity. A battery of psychophysiological signals was collected for two levels of cognitive workload from which 43 measures were derived. Three feature-reduction methods, principal component analysis, a weight-based partial derivative method, and a weight-based signal-to-noise ratio were applied, and the results were used as inputs to an artificial neural network for training and classification. Average classification accuracies up to 89.7 percent were achieved and the number of input features required was reduced by up to 84 percent.

14. SUBJECT TERMS

Artificial neural networks, Cognitive workload, Feature saliency, Psychophysiological measures, Principal component analysis, Nonlinear partial derivative, Signal-to-noise ratio

15. NUMBER OF PAGES

36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The primary motivation of this research is to provide real-time human cognitive state estimation and apply it to adaptive decision aiding in complex task environments. This paper investigates one of many barriers to real-time classification of operator state. Specifically, it is necessary to identify the proper features used by the classifier model.

Classification of operator mental workload has numerous applications in the fields of human factors engineering, training, testing, and evaluation. For example, knowledge of a pilot's state in an advanced fighter aircraft could be used to increase system efficiency and effectiveness by using this information as real-time guidance for an adaptive control system. In-flight cognitive load is merely one concern of USAF researchers. Uninhabited Air Vehicle (UAV) and Uninhabited Combat Air Vehicle (UCAV) operators may experience performance degradation during mission segments with high cognitive load. In addition, an understanding of cognitive workload could aid in development of human-computer interfaces by providing metrics of operator state. Accurate and reliable assessment of operator state is the key to successful implementation of adaptive automation or design evaluation. Several approaches have been applied to this problem, such as Baysian estimation and linear statistical techniques; none of which have achieved the required accuracy or reliability required for implementation.

Neural networks have several potential advantages that make them attractive for use as classifiers of operator state. They are adaptive and nonlinear, and they have the ability to generalize. Because of the inherent nonlinearity and the complex interactions among the features of cognitive activity during highly dynamic multiple task situations, accurate workload classification is difficult. In addition, the relationships between physiological

variables and performance are not well understood; therefore, the relevant features for classification are not known. Consequently, adaptive neural networks are an ideal choice for classifying mental workload in complex, real-world situations.

Neural networks have been used in classification of cognitive workload in several studies. Anderson, Devulapalli, and Stolz (1995) investigated single task workload classification using alpha band activity and autoregressive methods. Gevins, Smith, Leong, McEvoy, Whitfield, Du and Rush (1998), using EEG and artificial neural network classifiers, manipulated low, moderate, and high working memory load states and compared each load pair in the classification process. Cognitive workload estimation was investigated using EEG band activity and neural networks during a simulated landing task (Russell, Monett and Wilson, 1996, Greene, Bauer, Kabrisky, Rogers, Russell and Wilson, 1996), during simulated air traffic control (Russell and Wilson, 1998), and in an air to ground Scud hunt mission (Russell, Reid and Vidulich, 2000).

An important consideration in classification is determining the input features. This feature selection is essential for any classification problem or algorithm, be it nonlinear (neural networks) or linear (stepwise discriminant analysis). Some input features may be redundant because they are highly correlated or duplicated with only scalar differences. Others may fail to provide any useful information for discrimination. Decreasing the number of input features by removing redundant or meaningless inputs reduces the computation required for training.

Reducing the number of features also reduces the number of exemplars or samples necessary for adequate learning by the classification algorithm. The "curse of dimensionality" abounds in pattern classification problems such as cognitive load state

estimation. The psychophysiological signals, such as electroencephalogram (EEG), electo-oculogram (EOG), and electrocardiogram (ECG), collected in this study produce a gamut of derived features. As the number of input features increases, so do the number of training examples necessary to estimate the free parameters of the model.

This paper investigates three methods of input feature reduction. Principal component analysis (PCA), the Ruck weight-based partial derivative method, and a weight-based signal-to-noise ratio (SNR) method are compared using two types of derived input features (power variables and dimensionless variables). Input features selected by each method are presented to a multilayer perceptron artificial neural network for classification of two states of cognitive load.

METHODS

Subjects

Data from five naïve participants (designated as s01, s02, s03, s04, and s05) were collected, with all participants completing an approximately hour-long scenario for each of two days of data collection following familiarization training. Participants were paid for their participation.

The Multi-Attribute Task Battery Crewmember Simulator

The Multi-Attribute Task Battery (MATB) interactive software developed by NASA was used in this experiment (Figure 1). The MATB simulates tasks analogous to those a flight crewmember would encounter (Comstock and Arnegard, 1992). Tasks included monitoring, tracking, communication, and resource allocation responsibilities in a continually changing environment. These represented the same tasks performed by a UAV or UCAV operator. Each subject trained on MATB for several days until a consistent level of proficiency was achieved. Proficiency was declared when the performance parameters asymptote to minimum errors. This procedure helped to reduce potential learning effects and allowed subjects to achieve a desired level of familiarity and comfort with the laboratory setting.

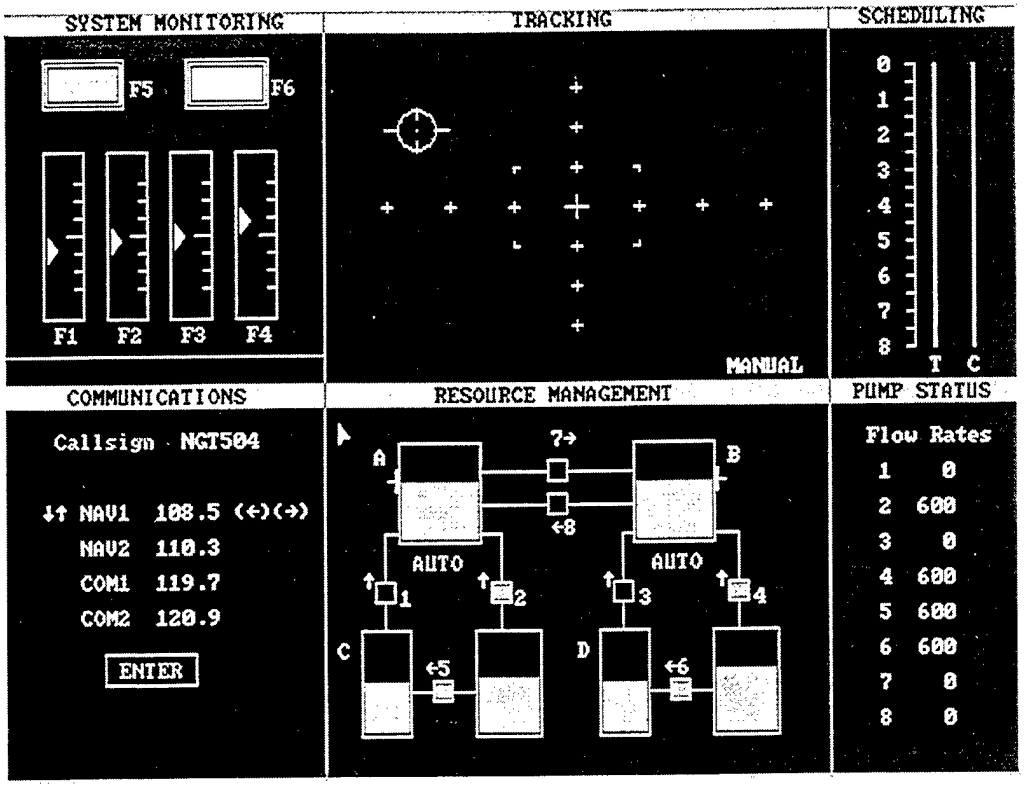


Figure 1. Sample MATB simulation display

Data Collection

Eight channels of EEG data were recorded at sites positioned according to the International 10-20 electrode system (Jasper, 1958). Mastoids were used as references. Electrode impedances were below 5K ohms. Each EEG channel was corrected for eye movement and blinks and the frequency spectrum was calculated and stored at each one-second interval. Eye blink, heart, and respiration intervals were also collected.

During each recorded scenario, subjects were presented a randomized sequence of low and overload cognitive workload levels. Two data collection runs were designated as training data for the classifier and consisted of 10 minutes of low and high workload (five minutes at each level). Two additional runs of data collection were designated as validation and testing sets for the classifiers and consisted of 15 minutes of alternating low and high cognitive load during both days of data collection. The experiment

monitored and recorded performance measures of required MATB tasks and psychophysiological data from each test subject.

PROCEDURE

Feature Extraction

Feature extraction is the processing of raw data into sets of measures that quantify a group of states for classification. It provides no additional information to the classifier algorithm; theoretically, using the raw data for the classifier would provide the best results. Due to the quantities of data, the required dimensionality of the artificial neural network (or indeed any classifier) makes using the raw data impractical. In practice we must develop a set of features that are manageable and reliable and that produce desired accuracies. Power of the EEG and intervals of the peripheral physiological measurements were used.

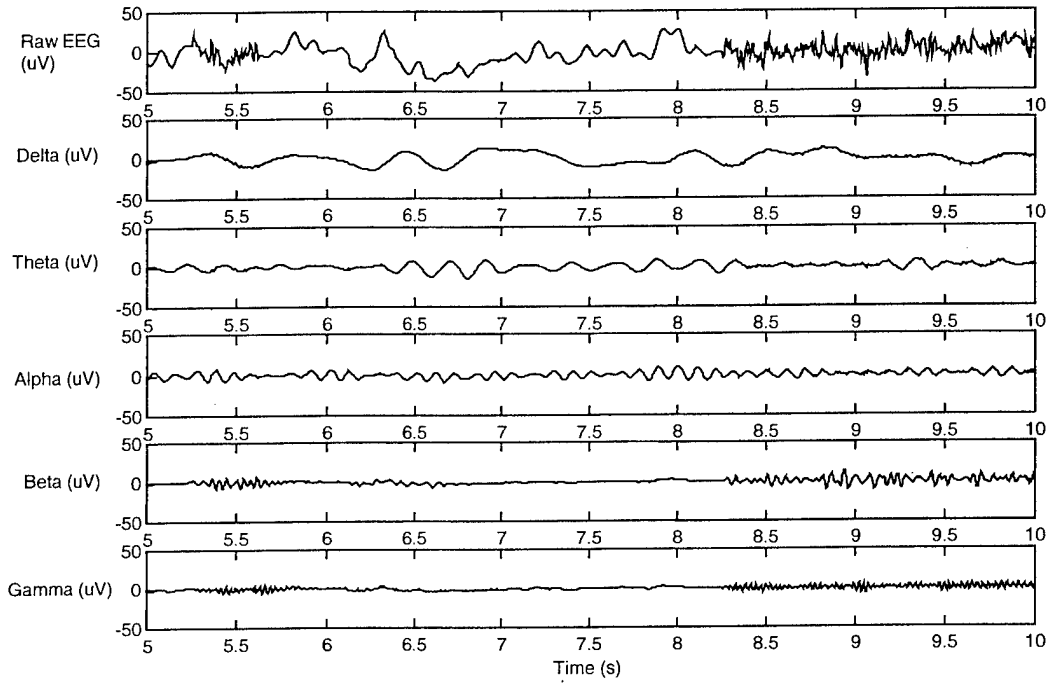


Figure 2. Sample EEG recording and traditional EEG bands

The frequency spectrum for each one-second interval was separated into the five traditional bands of EEG: delta (DC-3 Hz), theta (4-7 Hz), alpha (8-12 Hz), beta (13-30 Hz) and gamma (31-42 Hz). The time series representations of these bands are shown in Figure 2.

The data were segmented into forty-second windows with a 35-second overlap as shown in Figure 3. Parseval's Theorem states that the integral of the magnitude square of a time series is equal to the integral of the magnitude square of the time series Fourier coefficients. In other words, the energy in the time domain is equivalent to the energy in the frequency domain. Making use of this theorem, we determined the log power of each band using

$$P = 10 * \log\left(\sum F(\omega)^2\right). \quad (1)$$

Log power of delta, theta, alpha, beta and gamma from the eight sites were used, resulting in 40 features as inputs to the neural network. Three physiologically based features, the interval between eye blinks, interbeat intervals, and the interval between breaths, were also used as input features, resulting in 43 inputs.

Dimensionless features are recommended by most pattern recognition texts (Duda, Hart and Stork, 2001). Power ratios for the EEG bands and intervals relative to a resting baseline for heart, eye and respiration were calculated and used as inputs to the classifier. The power ratios were calculated for each EEG band with respect to the total power of the EEG spectrum. The interblink, interbeat, and interbreath intervals were adjusted relative to an average resting baseline value. These manipulations provided 43 dimensionless features to be used by a classifier.

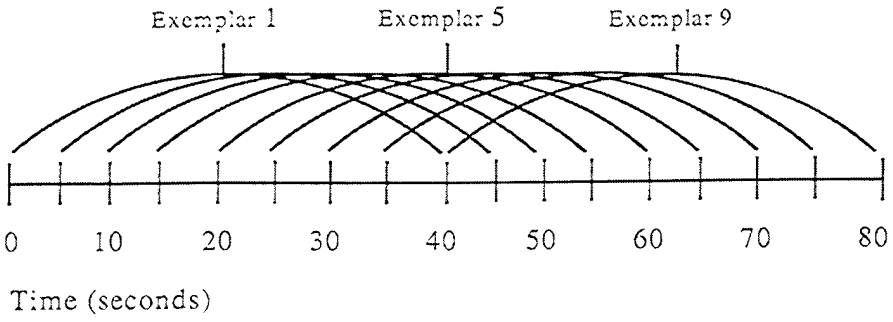


Figure 3. Description of moving window

Artificial Neural Network

A feedforward backpropagation neural network was used in this study (Widrow and Lehr, 1990; Lippmann, 1987). A backpropagation neural network classifier maps input vectors to output vectors in two phases. First, the network learns the input-output classification from a set of training vectors. Then, after training, the network acts as a classifier for new vectors.

The backpropagation algorithm initializes the network with a random set of weights for each fully connected layer, then the network trains using the input-output pairs.

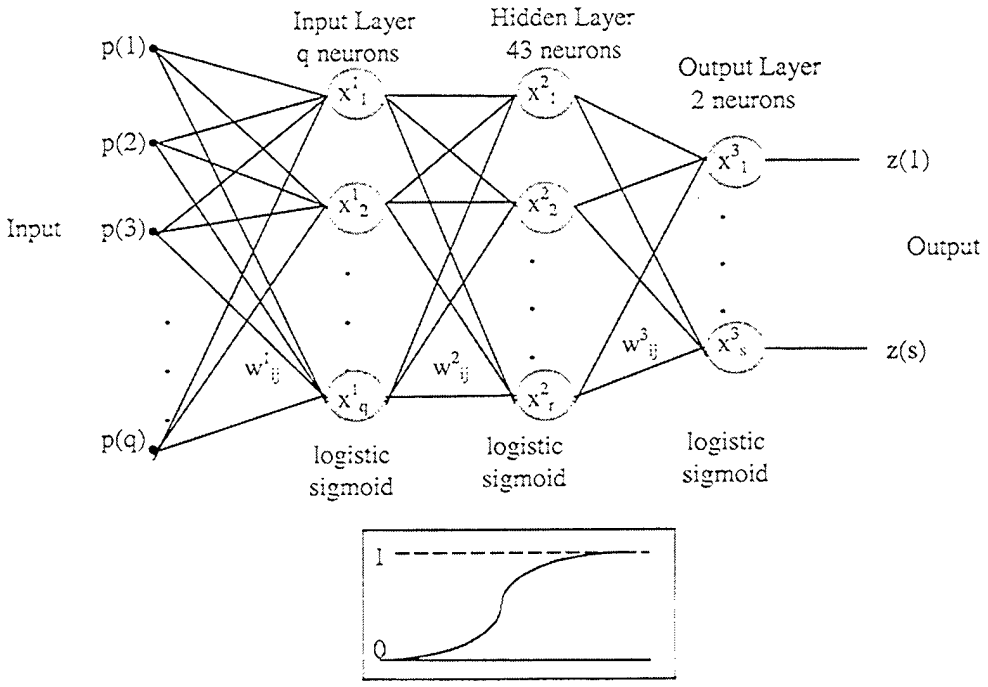


Figure 4. Network architecture showing a fully connected network with the number of neurons in each layer. The form of the logistic sigmoid activation function is provided.

The learning algorithm uses a two-stage process for each pair: forward pass and backward pass. The forward pass propagates the input vector through the network until it reaches the output layer. First, the input vector propagates to the hidden units. Each hidden unit calculates the weighted sum of the input vector and its associated interconnection weights. Each hidden unit uses the weighted sum to calculate its activation. Next, hidden unit activation propagates to the output layer. Each node in the output layer calculates its weighted sum and activation. Figure 4 shows the forward pass

and Figure 5 is a typical unit featuring the summation and the activation. The output of the network is compared to the expected output of the input-output pairs; their difference defines the output error. In the second stage of network training, the output error propagates backward to update the network weights. First, the error passes from the output layer to the hidden layer updating output weights. Next, each hidden unit calculates an error based on the error from each output unit. The error from the hidden units updates the input weights. One training epoch passes when the network processes all the input-output pair in the training set. Training stops when the sum-squared error is acceptable or when a predefined number of epochs is executed.

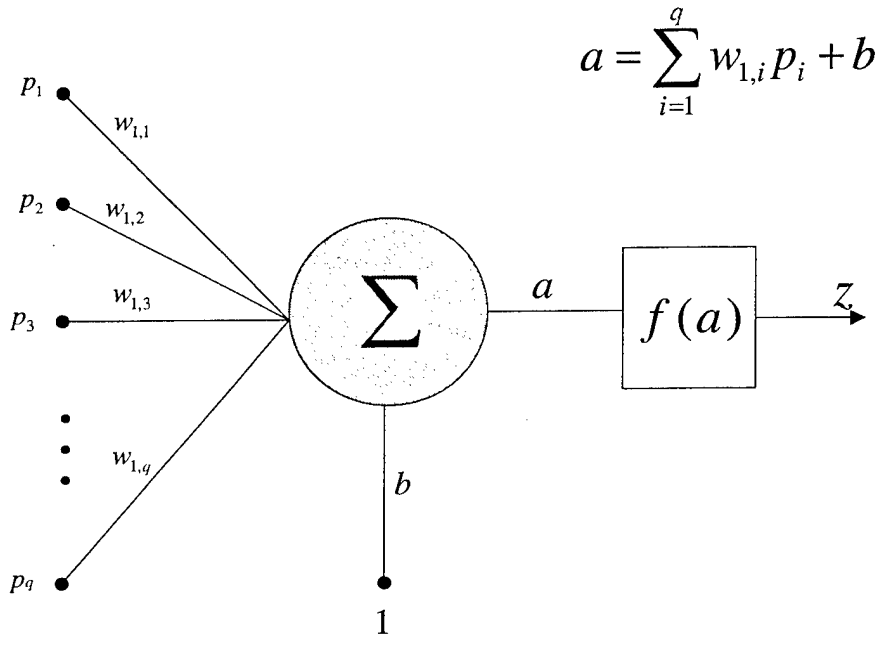


Figure 5. Individual neuron showing the weighted sum of the inputs followed by the logistic sigmoid activation function, *f(a)*.

The algorithm (backward pass) attempts to minimize the error or energy function defined

$$E = \sum_{i=1}^{m} \left\| \vec{z}_i - \vec{t}_i \right\|^2 , \tag{2}$$

where m is the size of the training set, z is the neural network output vector, and t is the expected output for each training input-output pair i.

It may be simpler to examine the algorithm as a series of steps. The steps for implementing a backpropagation neural network are as follows (Lippmann, 1987):

(1) Initialize the weights (w_i) and biases (b_i) where i is the current iteration.

(2) Present the input matrix (p) and the target vector (t).

(3) Calculate the output of the network (z_i).

(4) Calculate the error ($e = z_i - t$).

(5) Determine the new weights (w_{i+1}) where $i+1$ is the next iteration.

(6) Determine the new learning rate.

(7) Repeat steps 2 through 5 until desired error is achieved.

Mathematically, these steps were as follows: (Haykin, 1999; Widrow and Stearns, 1985; Widrow and Lehr, 1990). The weights and biases were initialized using a random number generator and limiting the values to the range –0.5 to 0.5, which is the nearly linear region of the hyperbolic sigmoid activation function.

The input data were normalized to zero mean and unit standard deviation using

$$pn(i) = \frac{p(i) - \mu}{\sigma} , \tag{3}$$

where pn is the normalized input vector, p is the input vector, μ and σ are the mean and standard deviation for each feature, and i represents the i^{th} exemplar. The target vectors were assigned based on the *a priori* target output class. The class target output neuron

was assigned 0.9 and the other target output neuron was assigned 0.1. The target vectors were [0.9 0.1] for low workload and [0.1 0.9] for the high workload condition.

The output of the network is determined by propagating the normalized input through each layer of the backpropagation neural network. It is necessary to examine the output of an individual neuron and then expand that understanding to the framework of the entire network. As shown in Figure 5, the output of the individual node or neuron is

$$z = f(a) \tag{4}$$

and

$$a = \sum_{j=1}^{q} \left(w_{1j} p_j + b \right), \tag{5}$$

where w_{1j} is the weight, p_j is the input, and b is the bias and $f(a)$ is the activation function acting on a. The figure suggests this neuron is in the input layer since the leading index on the weight is 1. Generalizing to any neuron results in

$$z_j = f(a_j) \tag{6}$$

and

$$a_j = \sum_{j=1}^{q} \left(w_{ij} p_j + b_j \right). \tag{7}$$

Activation functions can be linear or nonlinear. A common activation function is a sigmoidal nonlinearity. In our case, it is a logistic sigmoid function with an output range $0 \le f(a) \le 1$ in the form

$$f(a) = \frac{1}{1 + e^{-a}}. \tag{8}$$

The error is simply the difference between the output of the network and the expected target value:

$$E_k = \sum_{i=1}^{s} (z_i - t_i)^2, \tag{9}$$

where k is the error for the current input exemplar.

We can adjust the weights and try to minimize the error E_k through the backward path. Although the activation function is nonlinear, it is differentiable and we can compute $\frac{\partial E_k}{\partial w_{ij}}$, which we will make use of in our selection of a learning rule. The network algorithm is an extension of the Widrow-Hoff learning rule (Widrow and Lehr, 1990), which is a gradient descent algorithm based on Widrow's earlier work in Adaline and Madaline neural networks. This rule adjusts the weights using a steepest descent algorithm,

$$w_{ij}(n) = w_{ij}(n-1) - \mu \frac{\partial E}{\partial w_{ij}}, \tag{10}$$

where μ is a constant that controls the speed of convergence (learning rate).

Adaptive learning and momentum were used to decrease the time required for training the networks and to ensure the network reaches a global mimina. Typically, gradient descent methods use a fixed learning rate to control the rate of convergence. However, it is difficult to determine an optimum rate. If the fixed learning rate is too large, the gradient descent algorithm becomes unstable due to oscillations. If the learning rate is too small, the incremental steps along the error surface are small and in turn the algorithm takes a long time to converge to the desired error. Adapting the learning rate to optimize the learning progress can maintain stability while keeping the learning rate as large as possible to improve the rate of convergence. As the slope of the local error surface increases, the learning rate decreases to control stability.

Momentum prevents the network algorithm from becoming trapped in a local minimum. Essentially the algorithm will "jump over" or ignore small perturbations in the error surface. Modification of the delta-learning rule to include momentum results in a new learning rule

$$w_{ij}(n) = \alpha w_{ij}(n-1) - \mu \frac{\partial E}{\partial w_{ij}}, \tag{11}$$

where α is the momentum and μ is the learning rate.

This process is repeated until a desired error is achieved. The desired error is problem specific and must be determined. We determined our target or desired error by the validation method. The neural nets were optimized by a validation method. The data were segmented into three data sets: a training data set, a validation set, and a test data set. During training, the neural network adjusted the weights and biases based on the training data set. After each adjustment the weights were tested on the validation set and once the network reached a minimum solution the test set was used to evaluate the final weights. The training and the validation error initially follow the same path until at some point the neural network begins to learn the idiosyncrasies of the training data set. The error for the training data continues to decrease after this point, but the validation error increases due to the neural network overlearning the training data. The ideal stopping point for training the neural network is the minimum validation error.

Once trained, network weights are fixed and the net acts as a pattern classifier. As a classifier, the network examines input vectors it has never seen and predicts the class of the input vector.

The number of nodes in the input layer, the hidden layer and the output layer defines the neural network used in this study (See Figure 4). The number of input units

and the number of output units are problem dependent. Initially, in our case, the input layer consists of 43 neurons representing the 43 features that form the full input space. The output layer consisted of two neurons since the number of classes existing in the data determined the size of the output layer. The number of hidden units required is usually not known. Hidden units are the key to network learning and force the network to develop its own internal representation of the input space. The network that produces the best classification with the fewest units is selected as the best topology. A net with too few hidden units cannot learn the mapping to the required accuracy since the smaller hidden layer would limit interaction of the input space. Too many hidden units allow the net to 'memorize' the training data and will not generalize well to new data. We used 43 neurons in the hidden layer.

After completion of the feature reduction, twenty neural networks were trained randomizing both the training data order and initial weights. These neural networks maintained the same architecture described above with the exception of the number of neurons in the input layer. In each case the number of neurons in the input layer represented the number of salient features determined after the feature reduction methods were applied.

Principal Component Analysis

Principal component analysis (Jolliffe, 1986 and Flury, 1988) was used to reduce the number of input features presented to the artificial neural network classifier. PCA is a useful technique for multivariate analysis that can 1) transform correlated variables into uncorrelated variables, 2) determine linear combinations that have the maximum range of

variability, and 3) reduce data. We take advantage of all three properties in this study, but primarily the PCA was used for data reduction.

PCA projects the data on the direction of each of the eigenvectors determined by the eigenvalues of the characteristic polynomial of the covariance matrix. First the covariance matrix is determined. The characteristic polynomial of the covariance matrix, C, is determined by

$$|C - \lambda I| = \lambda^d + a_1\lambda^{d-1} + \cdots + a_{d-1}\lambda + a_d = 0, \tag{12}$$

where I is an identity matrix of the same order d as the covariance matrix C. The roots λ of the characteristic polynomial are the eigenvalues, and each eigenvalue has an associated eigenvector. The eigenvalues are ordered by size from the largest to the smallest and become the principal components of the covariance matrix.

The largest principal component is the eigenvalue that accounts for the largest variance of the covariance matrix. Therefore, the first (largest) principal component is the projection on the direction in which the variance of the projection is maximized. Selecting only the largest eigenvalues and their associated eigenvectors reduces the input space. The number of eigenvalues used in this study is the number of eigenvalues that explain more than 80% of the cumulative variance of the covariance matrix. The input features to the classifier are the 43 derived features projected onto those eigenvectors whose eigenvalues account for more than 80% of the explained variance. Therefore, the input features are linear combinations of the derived features weighted by the scalar components of the eigenvector or the factor loadings of the significant principal components.

Partial Derivative Method

The Ruck saliency measure (Ruck, Rogers and Kabrisky, 1990) was used to determine which features provide information for the classification algorithm. This technique calculates the partial derivative of each layer and rank orders the features based on the saliency measure. In essence, this method provides an input-output relationship between the network output layer and the input features. This partial derivative method is possible because although the activation function is nonlinear, it is differentiable. The derivative of the activation function (equation 8) used in this study is

$$f'(a) = f(a)(1 - f(a)). \tag{13}$$

Feature saliency is based on the concept that a fully trained network contains all the information for describing the relative importance or saliency of each of the input features. The partial derivatives look cumbersome but can be readily calculated using the chain rule and are easily implemented in vector form. These calculations are performed starting with the output layer. The partial derivative for the output layer is

$$\gamma_{k3}^3 = f'(a_{k3}^3) \tag{14}$$

$$= a_{k3}^3(1 - a_{k3}^3), \tag{15}$$

where k3 represents each output neuron and, in our case, the output layer is the third layer. Recall from equation 7 that a represents the weighted sum of the inputs to the activation function plus the bias or threshold. The second or hidden layer is more complex:

$$\gamma_{k2}^2 = f'(a_{k2}^2)\sum_{k2}\gamma_{k2}^3 w_{k2}^3 \tag{16}$$

$$= a_{k2}^2(1 - a_{k2}^2)\sum_{k2}\gamma_{k2}^3 w_{k2}^3 . \tag{17}$$

In this case, k2 represents the second layer neurons. The input layer has the same form as the second or hidden layer:

$$\gamma_{k1}^{1} = f'(a_{k1}^{1})\sum_{k1}\gamma_{k1}^{2}w_{k1}^{2} \tag{18}$$

$$= a_{k1}^{1}(1-a_{k1}^{1})\sum_{k1}\gamma_{k1}^{2}w_{k1}^{2}\,. \tag{19}$$

Finally the partial derivative for the entire neural network is

$$\frac{\partial z_j}{\partial x_i} = \sum_{k1}\gamma_{k1}^{1}w_{i}^{1}\,. \tag{20}$$

Combining equations 13 through 19 yields

$$\frac{\partial z_j}{\partial x_i} = \sum_{k1}\left[a_{k1}^{1}(1-a_{k1}^{1})\sum_{k1}\left[a_{k2}^{2}(1-a_{k2}^{2})\sum_{k2}\left[a_{k3}^{3}(1-a_{k3}^{3})\right]w_{k2}^{3}\right]w_{k1}^{2}\right]w_{i}^{1}\,. \tag{21}$$

Once the partial derivatives have been calculated the saliency can be determined for each feature as

$$\Gamma_i = \sum_{p}\sum_{j}\left|\frac{\partial z_j}{\partial x_i}\right|, \tag{22}$$

where Γ_i is the saliency for the *i*th feature, *j* ranges over the outputs and *p* ranges over the exemplar vectors in the training set.

Feature reduction was accomplished by an iterative approach. A network was trained using all the features described in the feature selection portion of this paper. The partial derivative saliency was calculated for each feature. The features were then rank ordered based on the computed saliency. The least salient feature was removed from the input matrix and the networks were retrained using the reduced feature set. This sequence was repeated until the only one feature remained. The minimum data set is the smallest set that has the highest classification accuracy.

Signal-to-Noise Ratio Method

The SNR saliency measure was used to quantitatively assess the saliency of each feature (Bauer, Alsing and Greene, 2001). This measure compares the input layer weights of an individual input feature to the weights of an injected noise feature. Theoretically, the measure will be significantly larger than zero for salient features and close to or less than zero for insignificant or nonsalient features. This method uses a fully trained network to determine the importance or saliency of each of the input features to the artificial neural network. The weights of each feature are evaluated with respect to a random noise variable injected into the artificial neural network. The features with high signal-to-noise ratio provide more information for classifying the target patterns than those with lower signal-to-noise ratios. The features with a negative signal-to-noise ratio provide little or no information for pattern classification since the signal-to-noise ratio of the injected random noise is zero. The SNR saliency metric is

$$SNR_i = 10 \cdot \log_{10}\left(\frac{\sum_{j=1}^{J}\left(w_{i,j}^{1}\right)^2}{\sum_{j=1}^{J}\left(w_{N,j}^{1}\right)^2}\right), \tag{23}$$

where SNR_i is the value of the saliency metric for feature i, j is the number of hidden nodes, $w^1_{i,j}$ is the weight from node i to node j, and $w_{N,j}$ is the first layer weight from the noise node N to node j. The injected noise has a Uniform (0,1) distribution.

The SNR metric can be used to rank order input features. If a given feature is not relevant to a neural network output, the updates of the first layer weights from the node of that feature should be random and fluctuate close to zero. On the other hand, if a given feature is relevant, the weights should be adjusted away from zero until error in the

network is minimized. Thus, the resulting SNR saliency metric should be significantly larger than 0.0 for salient features and close to 0.0 for non-salient features. The SNR saliency metric allows the comparison of the saliency of each feature to that of a baseline noise feature. This comparison, in turn, allows the SNR metric to be calculated and used at any time during network training.

The steps for implementing the signal-to-noise ratio saliency screening method is as follows:

(1) Introduce a noise (Uniform(0,1)) feature p_N into the feature set.

(2) Normalize all features.

(3) Initialize the weights and biases.

(4) Select training, validation and test sets.

(5) Initiate backpropagation training algorithm.

(6) Terminate training upon weight stabilization.

(7) Compute classification accuracy of the test data set.

(8) Compute SNR for each feature.

(9) Remove the feature with lowest SNR.

(10) Repeat steps 5 through 9 until all features are removed.

(11) Determine the smallest set of features with the highest classification.

RESULTS

Initial interpretation of the principal component analysis applied to the power variables revealed day-to-day variations of the data. Figure 6 shows a plot of the first two principal components (factors) for a subject. The data are clustering by day as well as by operator state, which was observed for most of the subjects. In an attempt to eliminate the day-to-day variation of the psychophysiological data, new dimensionless input features were derived as described in the feature extraction section above.

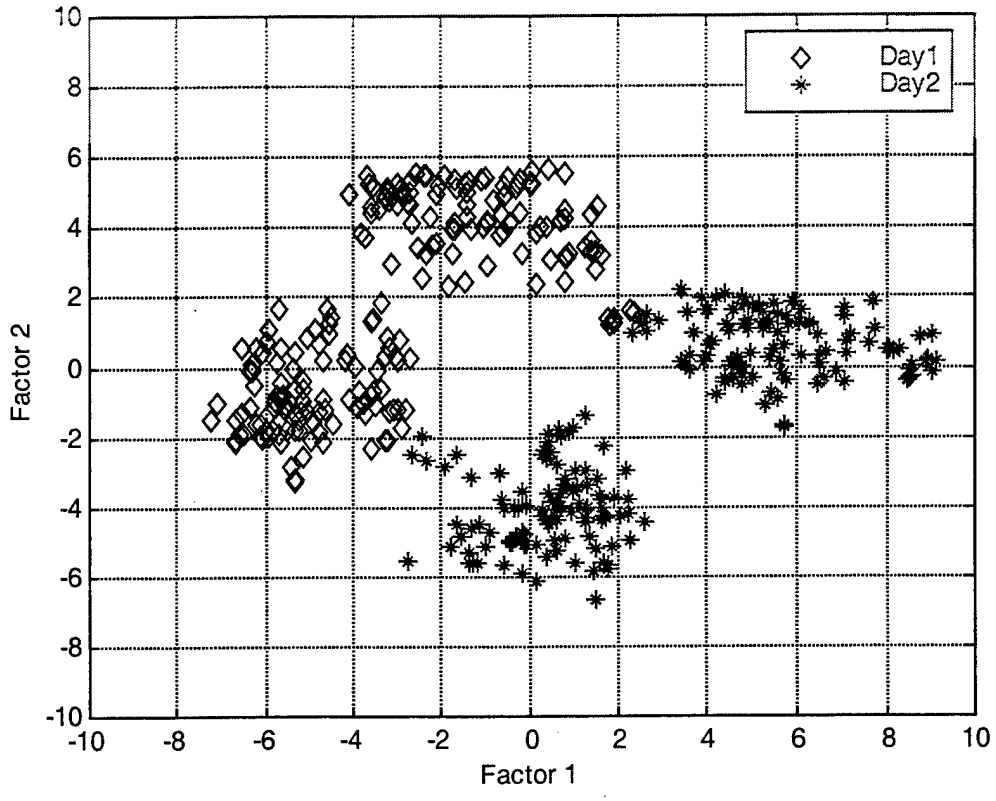
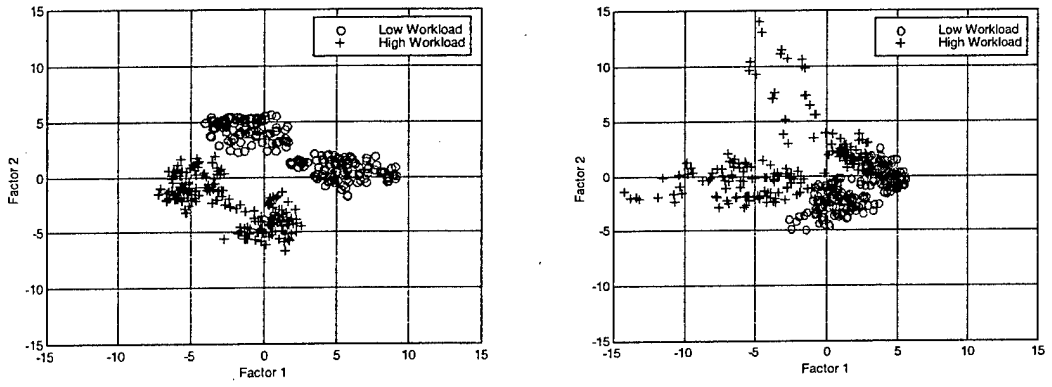


Figure 6. Sample subject showing day-to-day variations in the power input variables.

Using dimensionless features did produce the desired result of minimizing or eliminating the day-to-day variation. However, while reducing the day-to-day variation, the separation of the two cognitive states was also reduced (see Figure 7). Classification accuracy suffered by over ten percent as shown in Table 1.

Not only day-to-day variability but also subject-to-subject variability was investigated. Previous work (Russell and Wilson, 1998, Russell, Reid and Vidulich, 2000) indicate that individual classifiers must be developed for each participant due to the variability of the input features between participants as well as the number and identity of salient features. Figure 8 is a plot showing the clustering of the first two principal components by subject. The components were computed using the covariance matrix of the combined training data from all five subjects.



(a) (b)

Figure 7. Sample subject comparing a) power input variables and b) dimensionless input variables.

Mean classification accuracies were submitted to a 2 (*input variable type*) X 3 (*feature selection method*) X 20 (*repetition*) repeated measures ANOVA. Significant sources of variance resulting from the analysis included a main effect of *input variable type*, $F = 10.09$, $p < 0.05$ and a significant *input variable type* X *feature reduction method* interaction, $F = 4.71$, $p < 0.05$. No significant main effects of *feature reduction method* or *repetition* were noted. Post hoc pairwise comparisons of the *feature reduction method* and *input variable type* revealed the power variable accuracies were independent of *feature reduction method.*

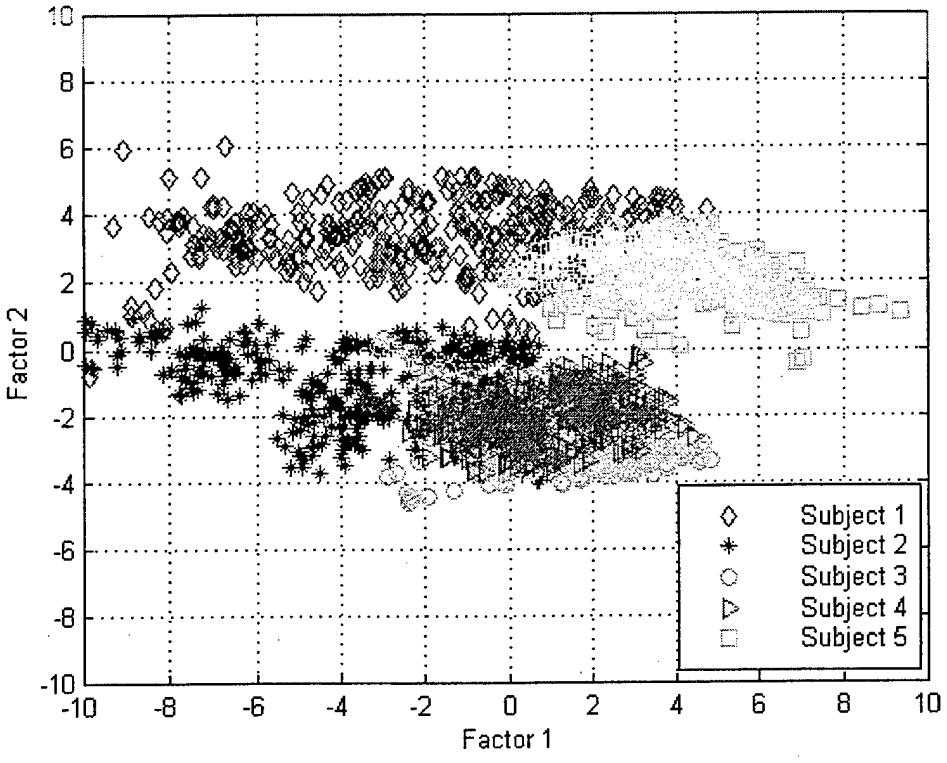


Figure 8. PCA projections indicating the data are clustering by subject.

Table 1. Classification accuracies by subject, input feature type and feature selection method.

	Dimensionless Variables			Power Variables		
Subject	PCA	Ruck	SNR	PCA	Ruck	SNR
1	86.9	91.7	87.7	93.5	89.7	86.5
2	84.5	93.4	79.4	95.0	95.7	97.3
3	82.8	79.6	66.7	98.0	96	96.5
4	67.8	71.0	59.9	71.0	71.1	78.7
5	73.3	83.4	77.1	85.2	92.6	89.5
Average	79.1	83.8	74.2	88.5	89.0	89.7

Table 2 shows the number of input features used by each method for both the power and dimensionless data. The Ruck weight-based partial derivative method with power variables required almost three times as many features to classify the cognitive state as did the PCA and SNR methods. The dimensionless variables required about the same number of features to classify the cognitive state. However, using power variables as inputs to the neural network classifier produced the same classification regardless of feature selection method. The Ruck partial derivative method produced the best results (83.8% correct classification) when using dimensionless variables.

Table 2. Number of input features used by input feature type and feature selection method.

	Power Variables	Dimensionless Variables
PCA	8	10
Ruck	22	9
SNR	7	13

The saliency values of each of the significant features were averaged across subject and rank ordered from highest to lowest saliency. The electrode site locations of the salient features for each of the feature saliency methods and input variable type are presented in Figure 9. The Ruck partial derivative method used five of the eight EEG

electrode sites in addition to eye blink activity and interbreath interval when using power variables as input features. The use of dimensionless data reduces this number to three of eight EEG sites and retains eye blink activity as salient sites. The signal-to-noise ratio method increased the number of psychophysiological measures when using dimensionless data. Six of eight electrode sites were used along with interbeat, interbreath, and interblink intervals.

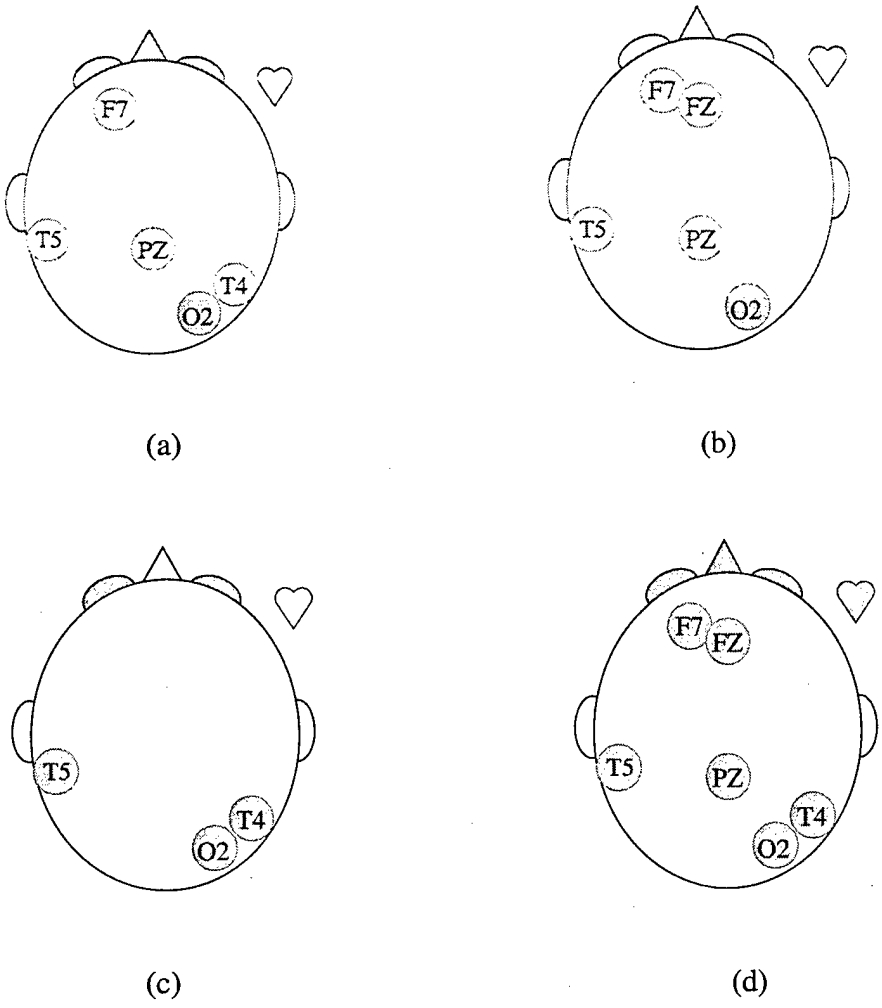


Figure 9. Salient feature selected by (a) Ruck power variables, (b) SNR power variables, (c) Ruck dimensionless variables and (d) SNR dimensionless variables.

The EEG frequency band most prevalently selected as salient was the beta band followed by the delta and gamma bands. These three bands represent the two extremes of the frequency spectra, specifically DC to 4 Hz and 13 to 42 HZ. The cognitive psychology literature typically associates alpha and theta EEG bands with changes in cognitive load. Both of these variables were selected with the least frequency by all feature selection methods used in this study.

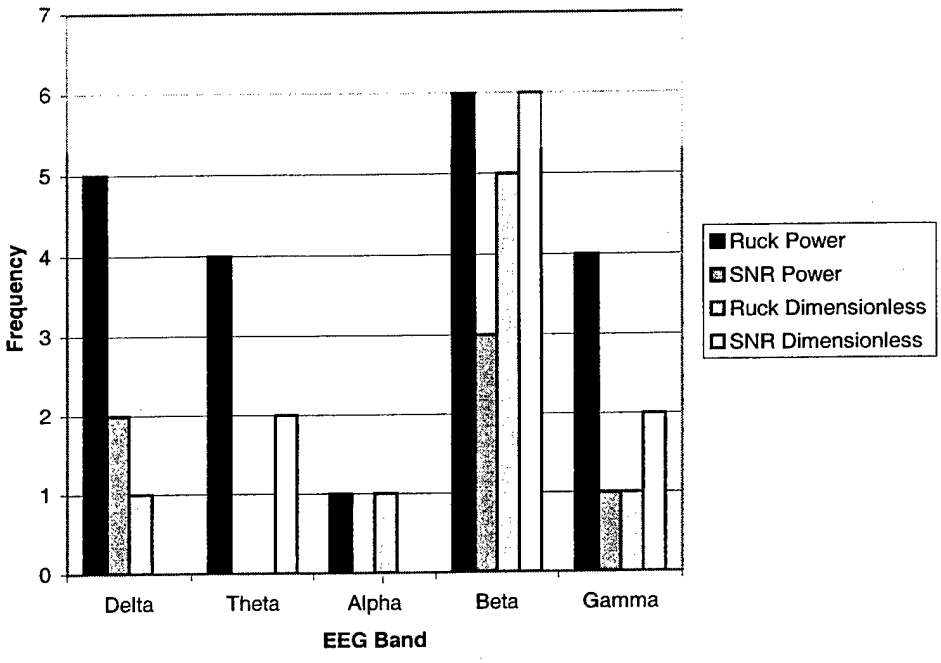


Figure 10. Frequency of EEG band selection associated with variable type and feature selection method.

DISCUSSION

Feature reduction can be accomplished, resulting in a significant decrease in the dimensionality of the neural network model. This reduction not only decreases the amount of training data required by the model but also reduces the training time for the multilayer perceptron. For this study, the selection of a feature reduction method has no bearing on the level of classification accuracy when using the derived power variables.

This study indicates that principal component analysis is probably a better choice of input feature selection method for a number of reasons. The partial derivative method and the signal-to-noise ratio method selected different subsets of the total input space for each individual participant. The number and identity of salient input features varied considerably not only between participants but additionally varied between methods of feature saliency selection. This result indicates that the number of electrodes cannot be reduced, since the location and identity of electrode sites vary by subject. PCA does not require the removal of inputs, since this method uses a weighted linear combination of all measures and only the combinations that contribute to the majority of the explained variance are used.

Another advantage in using PCA for input feature reduction is it's computational efficiency. Both the weight-based partial derivative method and the signal-to-noise ratio method require the training of multiple neural networks to determine feature saliency and removal of nonsalient features. Increasing the number of input features increases the number of neural networks to be trained. PCA does not require an artificial neural network for selecting salient features: input feature selection is based on those weighted

linear combinations of derived measures that account for the majority of the variance in the covariance matrix.

An additional advantage of PCA is the number of eigenvalues that explain 80% of the variance is robust. Eight weighted linear combinations of the power variable input space were required to explain 80% of the variance of the covariance matrix. This number varied by less than one eigenvalue across subjects. The Ruck partial derivative method used 22 input features that varied by as much as six features across subjects. The SNR method required seven features and that number varied by as much as eight features across subjects.

The very principles that create advantages for the PCA approach produce a number of disadvantages. A disadvantage to principal component analysis is that this method does not directly reduce the quantity of data that must be collected. Both the weight-based partial derivative method and the signal-to-noise ratio method will reduce the number of electrodes applied to the participant. Another disadvantage to PCA is there is no direct relationship between the cognitive state and the frequency bands of each individual electrode site. For example, it is much more difficult to determine if a significant factor in determining cognitive load is an increase in T5 beta power.

REFERENCES

Anderson, C.W., Devulapalli, S. and Stolz, E.A. (1995). Determining Mental State from EEG Signals using Parallel Implementations of Neural Networks, *Scientific Programming, Special Issue on Applications Analysis*, Vol. 4, No. 3, pp. 171-183.

Bauer, K.W., Alsing, S. G., and Greene, K.A. (To appear 2001). "Feature Screening using Signal to-Noise Ratios," *Neurocomputing*, accepted April 29, 1999.

Comstock, J. R., and Arnegard, R. J. (1992). The Multi-Attribute Task Battery for Human Operator and Strategic Behavior Research, NASA Technical Memorandum 104174.

Duda, R.O., Hart, P.E., and Stork, D.G. (2001). *Pattern Classification*, New York: John Wiley & Sons.

Flury, B. (1988). *Common Principal Components and Related Multivariate Models*, New York: John Wiley & Sons.

Gevins, A., Smith, M.E., Leong, H., McEvoy, L., Whitfield, S., Du, R., and Rush, G., (1998). Monitoring Working Memory Load During Computer-based Tasks with EEG Pattern Recognition Methods, *Human Factors*, vol. 40, no. 1, pp. 79-91.

Greene, K.A., Bauer, K.W, Kabrisky, M., Rogers, S.K., Russell, C.A., and Wilson, G.F. (1996). A preliminary investigation of selection of EEG and psychophysiological features for classifying pilot workload, In Dagli, et al. (Eds.), *Smart Engineering Systems: Neural Networks, Fuzzy Logic and Evolutionary Programming,* ASME Press, pp. 691-697.

Haykin, Simon (1999). *Neural Networks: A Comprehensive Foundation*, Upper Saddle River, NJ: Prentice Hall.

Jasper, H. H. (1958). The ten-twenty electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology*, vol. 10, pp. 371-375.

Jolliffe, I.T. (1986). *Principal Component Analysis*, New York: Springer-Verlag.

Lippmann, R. P. (1987). An Introduction to Computing with Neural Nets, *IEEE Acoustics, Speech, and Signal Processing Magazine*, 4, No. 2, pp. 4-22.

Ruck, D., Rogers, S., Kabrisky, M. (1990). Feature Selection using a Multilayer Perceptron, *Journal of Neural Network Computing*, 2, No. 2, pp. 40-48.

Russell, C.A., Monett, C.T. and Wilson, G.F. (1996). Mental Workload Classification Using a Backpropagation Neural Network. In Dagli, et al. (Eds.) *Smart Engineering*

Systems: Neural Networks, Fuzzy Logic and Evolutionary Programming, ASME Press, 685-690.

Russell, C.A., Reid, G.B., and Vidulich, M.A., (2000). Pilot Workload Classification Using Artificial Neural Networks in a Simulated Scud Hunt Mission. *Proceedings of the 44th Annual Meeting Human Factors and Ergonomics Society*, 3, 109.

Russell, C.A. and Wilson, G.F. (1998). Air Traffic Controller Functional State Classification Using Neural Networks, In Dagli, et al. (Eds.) *Smart Engineering Systems: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Rough Sets,* ASME Press, 649-654.

Widrow, B. and Lehr, M. (1990). 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation, *Proceedings of IEEE*, 78, no. 9, pp. 1415-1442.

Widrow, B. and Stearns, S.D. (1985). *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ.

Wilson, G.F., Lambert, J.D. & Russell, C.A. (2000). Performance Enhancement with Real-time Physiologically Controlled Adaptive Aiding, *Proceedings of the 44th Annual Meeting Human Factors and Ergonomics Society*, Vol. 3, pp. 61-64.