Automated Analysis of the Clock Drawing Test for Differential Diagnosis of Mild Cognitive Impairment and Alzheimer's Disease

Russell Binaco, Nicholas Calzaretto, Jacob Epifano
Sean McGuire, Muhammad Umer, and Robi Polikar
Signal Processing and Pattern Recognition Laboratory
Rowan University Glassboro, New Jersey
Email:

Sheina Emrani, Victor Wasserman, and David J. Libon
New Jersey Institute for Successful Aging
School of Osteopathic Medicine (SOM)
Rowan University, Stratford, New Jersey
Email:

***Abstract*—Early and accurate diagnosis of Alzheimer's disease is still an unsolved problem. Practitioners have been seeking a noninvasive diagnostic tool to assess a patient's level of cognitive impairment. The clock drawing test is one such tool, where patients are asked to draw an analog clock showing the time 10 past 11. Traditionally the clock test, combined with a battery of other neuropsychological tests, is evaluated by a neuropsychologist in order to reach a diagnosis. However, this is a time consuming as well as subjective process, with a significant rate of misdiagnosis. Researchers are now looking at machine learning algorithms to help form a relationship between the various features obtained from the clock drawing test and a degree of cognitive impairment. To do so, we extracted hundreds of features from the clock drawing test using a smart pen and tablet. We then used wrapper, embedded and information theoretic feature selection techniques to determine the most relevant features, and used them to train a neural network type classifier. The classifier is used to analyze these features and identify each patient as SCI (Subtle Cognitive Impairment), MCI (Mild Cognitive Impairment), or AD (Alzheimer's Disease). Preliminary results with a non-optimized classifier indicate a differential diagnosis accuracy of mid 70% to low 80% are achievable. These results, if further improved upon, can allow general health practitioners to use clock drawing test as a preliminary screening tool and recommend patients for further neuropsychological testing if warranted.**

I. Introduction

A. Medical Background

Alzheimer's disease (AD) is a major public health problem; by 2050 it is estimated that 14 million people may be affected by this illness [1]. In recent years much interest has been generated in the diagnosis and characterization of Mild Cognitive Impairment (MCI) syndrome which is believed to be a prodrome to AD as well as other dementia disorders. A variety of MCI subtypes have been identified including patients presenting with predominant amnestic (aMCI), dysexecutive (dMCI), and a combined or mixed (mxMCI) phenotype. Past research also suggests that each MCI subtype may eventually make the conversion to dementia [2]. As disease modifying treatment for AD becomes available identifying patients who are at risk for emergent MCI/dementia; and identifying patients early in the course of their illness will be particularly important.

The Clock Drawing Test (CDT) is a widely used neuropsychological test to differentiate between unaffected individuals, MCI subtype, and different dementia syndromes [3]. The CDT is used because of its ability to characterize dementia and MCI syndromes because of the large amount of data that is produced and the analysis of errors in the clock. One limitation of the CDT as a diagnostic tool is that the current scoring system is not able to capture small but significant features that could signal the presence of an early cognitive impairment.

The original CDT contains two testing conditions: command and copy. Both conditions ask the patient to draw an analog clock at a time "ten after eleven." The way the command is given forces the patient to decode the meaning and extract the real command. The command condition tests a patient's auditory comprehension, visual memory and abstraction. Before the drawing is executed, the verbal command must first be understood, and then the subject must recall the appropriate visuospatial image from memory. The copy condition, on the other hand, relies heavily on the visuospatial ability of the patient. Both of these conditions are important because the properties that each test do not overlap with one another. [19]

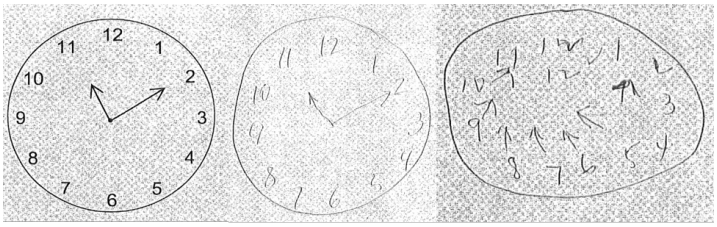

Figure 1. Example of Copy Clock Condition for two different patients

Recently, a digital version of the CDT (dCDT) [18] has been developed by Lahey Clinic and MIT in collaboration with the ClockSketch Consortium. The dCDT employs a digital pen to capture patients drawings. This device is able to store

over 350 different variables and each variable in the dataset corresponds to: a construction variable related to the how the clock is drawn, a time variable related to how long it takes to draw each construction variable, or a spatial variable related to where each construction variable is drawn. The solution to the aforementioned problem is solved and the data is now readily available to practitioners. In this paper we discuss the use of machine learning in conjunction with the dCDT to achieve higher than practitioner level classification accuracies given only clock drawing data.

B. Machine Learning Applied to dCDT

In recent years, machine learning has proved to be an invaluable tool in the medical field due to its capability to automate processes which are usually time-consuming and subjective, to get unbiased repeatable results [9]. Additionally, machine learning is extremely effective at handling big data that is difficult for humans to interpret. Medical technology has the ability to generate far more digital information than ever before allowing the research community to take advantage of machine learning techniques and their ability to tackle datasets with high dimensionality and a large amount of samples. Moreover, machine learning is helpful in solving issues such as human interpretability, lack of data availability, or unbalanced data [9]. The techniques that were used are explained in greater detail in the approach section.

To predict the cognitive state of a patient based on clock drawing data a strong classifier is needed. In the past others have attempted to classify cognitive impairments using the clock drawing test with gaussian SVM, random forests, CART, C4.5, boosted decision trees, and regularized logistic regression. This paper also covered the use of mRMR feature selection to use the top 200 features with different classifiers mentioned above [4]. Various types of neural networks such as feedforward neural networks and autoencoders have been used as well with promising results [15], [10].

C. Feature Selection applied to dCDT

In addition to a classifier, a proper feature selection algorithm is needed as well. This especially holds true for a high-dimensional dataset such as clock drawing data with over 350 features and only 163 instances. Many of these features are irrelevant, redundant, or noisy data and removing these types features has been shown to speed up processing, improve performance, and augment result comprehensibility [11]. Another benefit of feature selection, especially for the clock drawing data, is the ability to give a ranking to the most relevant features. This information can be very beneficial to doctors because it can give them understanding into what decision/memory thinking process during the clock drawing are predictors of cognitive impairment.

When it comes to feature selection there are three main categories of selection algorithms: wrapper methods, embedded methods, and filter methods. In wrapper methods, the objective function to derive the optimal amount of features is the classifier itself. An exhaustive search for the ideal combination of features using wrapper methods exponentially increases in computational complexity as the number of features increase. For this reason, algorithms such as sequential search, Genetic Algorithms and particle swarm optimization attempt to heuristically find an ideal set of features to maximize the performance of the classifier [6].

Embedded methods, such as LASSO and Elastic Net, also attempt to produce an objective function of the model. In embedded methods, the goal is to use regularization to find an objective function to fit the dataset. To regularize the objective function, a penalty factor based on the weights of feature coefficients is added to the cost along with the mean squared error of the function. In this way, an objective function can be formed that doesn't over-fit the data and as the feature coefficient weights decrease the most relevant features can be determined [16].

The last category of selection algorithms is filter methods. In filter methods, ranking criteria is used score the relevancy of each feature. In this way, a score is determined for each feature and the most relevant features are selected based on a threshold [6]. Different types of ranking criteria consist of correlation coefficient, variance, and information theory. For our applications, information theory was the chosen feature selected algorithm. In information theory, a score is given based on the mutual information between the label and the given feature. There are also several more advanced information theoretic criteria that take into account dependency between the feature and label as well as independence between features [5]. The concept of mutual information and the information criteria used is explained in more detail in the approach.

To represent wrapper, embedded, and filter selection approaches, a selection method was conducted representing each of these approaches for extracting the relevant clock data. The method chosen for wrapper approach was Recursive Feature Elimination with a Linear SVM kernel (SVM-RFE). This method was selected due to it's computationally quick speed. The method chosen for embedded approach was Elastic Net. Elastic Net is a combination of both Ridge and LASSO regression which makes the algorithm more robust in feature selection (this is explained in greater detail in the approach section). Finally, Information Theory was the selected filter approach because of its flexibility in practical applications (the multitude of criterion functions available) as well as quick computational speed. Another advantage of information theoretic methods, and feature ranking in general, is the fact that these methods do not rely on learning algorithms, which can be biased since the data is being changed to fit the learning algorithm [6].

D. Standards and Constraints

1) Standards: Through this project the following standards were utilized:

- Administration of Clock Drawing Test
- PEP 8 – Style Guide for Python [7]
- MATLAB Programming Style Guidelines [8]

2) Constraints: This project was constrained by the following:

- Size of data set (163 patients)
- Neural Network Size - Avoiding over fitting and under fitting the neural network to the data set based on the depth and width of the network
- Vectorization of data from pen into feature vector not all data from original drawing is capture ex drawing more than 12 digits.

II. Approach

A. Patient Acquisition

Participants studied in the current research were recruited from Rowan University's New Jersey Institute for Successful Aging, Memory Assessment Program (MAP). Patients were excluded if there was any history of head injury, substance abuse, or major psychiatric disorders. For all participants a knowledgeable family member was available to provide information regarding functional status. All MAP patients underwent a comprehensive neuropsychological evaluation. A clinical diagnosis was determined for each patient at an interdisciplinary team conference.

The dataset contained 163 patients total. Of the 163, they were split up into four different classes: SCI, MCI1, MCI2, and AD. These groups each contain, 35, 26, 43, and 59 patients respectively. The average age for each group is 77, 75, 75, and 79 respectively. 98% of the patients in the dataset are Caucasian/White and contain some level of post secondary education. 60% of the patients are females and 40% are males.

B. Engineering Design

The goal of this project was to differentiate different levels of cognitive decline in patients using the clock drawing test. To differentiate the classes we broke the problem into many binary problems, three class problems, and a four class problem. The four classes investigated are Subtle Cognitive Impairment (SCI), Amnestic MCI (MCI1), mixed MCI (MCI2) and AD.

When considering the entire dataset, four labels are present. This creates a four-class problem, but subdivisions of the dataset were considered for two- and three-class problems as well. Practically, while the four-class problem would be useful, the differentiation between healthy (SCI) and Alzheimer's Disorder (AD) is typically very easy to observe. It follows that three-class problems which consider the two MCI classes and only SCI or AD would be useful. Additionally, each of the six two-class problems (all pairwise combinations of two classes) were considered. The purpose of considering each binary case is to determine how separable each of the classes are, and to provide a way to eliminate uncertainty between two diagnoses. All of the test cases are listed explicitly as follows:
Two class problems: SCI vs MCI1, SCI vs MCI2, SCI vs AD, MCI1 vs MCI2, MCI1 vs AD, and MCI2 vs AD
Three class problems: SCI vs MCI1 vs MCI2, MCI1 vs MCI2 vs AD
Four class problem: SCI vs MCI1 vs MCI2 vs AD

In performing these experiments we applied information theory feature selection algorithms to the problem. Using each algorithm (MI, MRMR, JMI, CMIM) features were selected in intervals of 25 from 25 to 125.The feature selected data sets were run through the neural network architectures for each experiment. The neural network architectures tested were 1 layer 50 nodes, 2 layer 10 nodes each, and 2 layer 20 node first layer 10 node second layer. The networks were selected by selecting shallow smaller networks so that the network would not be able to over fit to the dataset. Smaller networks such as 1 layer 10 nodes were tested, but these extremely small networks did not have the capacity to achieve the accuracies which have shown from the larger networks. This lead to 540 results, for each network architecture and experiment the best performance (validation accuracy) with the lowest confidence interval was selected (see Table II). All networks were run with 10-fold cross validation, this means that the data was broken up into 10 different sets with 90% of the data for training and 10% of the data for validation, the 10 folds covers the entire data set, by averaging the validation accuracies across 10 folds it gives an excellent metric of performance on the dataset. It is important to break the data into training and testing groups to avoid over fitting a neural network to the data set. The training data is used to determine the weights of the model, and the validation data is used to determine the validity of the model.

C. Data Preprocessing

Before the raw clock drawing test data could be processed by feature selection algorithms, several different stages of preprocessing were required. First, there were some features that were clear duplicates of one another (i.e. features specifying what was drawn before a particular element of the clock, both with and without noise. In these cases, noisy features were omitted from the data). Additionally, there was a set of very sparse features all relating to the center dot. Since the center dot was rarely drawn and any individual who did not draw the dot did not have data for these features, only the feature specifying the number of strokes for the center dot was retained. Non-clock drawing data such as age was also removed.

All continuous features (such as time-based features, rather than categorical features that refer to specific components of the clock) were standardized to give all of the features the same range. To use the feature section algorithms all continuous features also had to be discretized; this was done by binning the data into 10 bins. The data was binned by using the discretization function in Gavin Brown's Matlab toolbox [5]. The process of binning takes continuous variables and divides them into N categorical values, where the N bins have equal ranges. This discretization was necessary in order to produce the probability density functions used by Brown's toolbox.

Another issue that had to be resolved before feature selection could be performed was dealing with missing data. Throughout the test, if patients omitted particular digits or hands from their clocks, those columns appeared missing in the dataset. For example, if a patient didn't write the

number three in the test, features like time to draw digit three, dimensions of digit three, etc. would be blank. In order to resolve this issue, a KNN search algorithm was used to fill in these missing sections. In KNN search, the algorithm found the k nearest neighbors (patients) that have the closest euclidean distance to features of the patient with the missing data. Once these k nearest neighbors were found, their features were averaged together and that number was filled in for the patient with missing data [14]. To find the K-nearest-neighbors, the euclidean distance is only calculated on features that are completely filled in as well as relevant to the missing features. For example, if a patient is missing data for the digit three features, a KNN search will be performed on features pertaining to digit one and two (assuming these are completely filled in). Its also assumed that the patient draws digit three similar to its KNNs based on digits one and two. For this reason, the average of the patient's KNN digit three features is calculated and filled in for the missing sections. To account for patients who forgot digits and or hands when drawing their clock, a binary variable was included where a one is entered if the digit was missing and zero is entered if it was written. In this way, there are no empty blocks in the data set while still taking into account missing features.

D. Feature Selection: Wrapper Approach

As previously mentioned, wrapper methods are exhaustive searching methods that use the performance of a classifier as the evaluation method for feature selection. The wrapper approach chosen was Recursive Feature Elimination with a Linear SVM Kernel (SVM-RFE). RFE is a sequential searching method that starts with the complete set of features in the dataset and sequentially eliminates features, one at a time. For each current set of selected features, the classifier is evaluated. Linear SVMs contain an explicitly computed weight vector which contains weights associated with each feature. These weights are used as a ranking criterion for RFE, where the lowest-ranking feature is removed [21]. This process is performed until every feature except one has been eliminated, and the feature set that produced the highest SVM classification performance is the optimal set of features.

Algorithm 1 Recursive Feature Elimination (RFE)

```
procedure RFE(features, labels)
    while features.length > 1 do
        SVM.fit(features,labels)
        SVM.evaluate(features, labels)
        w ← SVM.weights;
        sort features based on w;
        features ← features − features[−1];
    return optimalFeatures
```

Tables XIV to XXII include results for the SVM-RFE Wrapper method for each test case and each neural network architecture, and Table IV shows the overall best results from this method.

E. Feature Selection: Elastic Net Approach

The embedded approach used was Elastic Net. As explained in the introduction, Elastic Net attempts to produce an objective linear model to fit the data through regularization. In this way, an ideal linear model is created where the minimization of mean-squared-error as well as feature coefficients is taken into account when minimizing function cost. Elastic Net is a hybrid of LASSO and Ridge regression because it attempts to minimize both the L1 and L2 norm of the feature coefficients. As this happens, features drop out as coefficients drop to zero therefore becoming a feature selection algorithm. Moreover, Elastic Net has been seen as particularly useful when the number of predictors is much bigger than the number of observations (the clock drawing data has over 350 features but only 163 observations) [17]. By contrast, LASSO is not a very satisfactory variable selection method when $p >> n$ making Elastic Net the embedded approach of choice. Elastic Net equation is shown in Eq(1) and Eq(2).

$$\beta = min(\frac{1}{2N}\sum_{i=1}^{N}(y_i - x_i^T\beta)^2 + \lambda P_\alpha(\beta)) \tag{1}$$

$$P_\alpha(\beta) = \frac{1-\alpha}{2}||\beta||_2^2 + \alpha||\beta||_1 \tag{2}$$

In Eq(1) and Eq(2) y represents the class labels, x represents the features and β is the feature coefficients. α is a value between 0 and 1 and this value dictates the weights of the L2 norm vs the L1 norm during minimization. When α is 0, the L1 norm drops out of equation (Ridge regression) and when α is 1, the L2 norm drops out (LASSO regression). Finally, λ is a scalar value that increases the weight of the feature coefficients during minimization. If λ is very large, β is weighted heavily in the cost function and more β coefficients will drop out. Likewise, a small λ won't weigh βs in the cost function and the objective function would be calculated based on mean-squared error alone (βs will be large).

For the reason stated above, λ is the main parameter that dictates whether the objective function over-fits (λ too large) or under-fits (λ too small) to the data. To find the ideal λ for each test case 10-fold cross validation was used in Matlab while incrementally increasing λ at a set α. Through this method, the ideal λ was found for each test case, and features were selected based on the β coefficients that did not drop out at that λ. A pseudocode of this process can be seen in algorithm 2 along with a graphical representation in figure 2.

Algorithm 2 Elastic Net

procedure ELASTICNET(Y, X, α) ▷ Inputs: class labels (Y), features (X), alpha; Output: most relevant features
  λ = 0 ▷ initialize lambda at zero
  while $sum(|\beta| \neq 0)$ **do**
    for i in each fold (10-fold cross validation) **do**
      Xtrain, Ytrain ▷ training data
      Xtest, Ytest ▷ test data
      $\beta = min(f(\lambda, Xtrain, Ytrain, \alpha))$ ▷ calculate Beta that minimizes elastic net equation
      $MSE(i) = (Ytest - Xtest * \beta)^2$ ▷ compute MSE of test data
    $AvgMSE = mean(MSE)$ ▷ average MSE for CV
    $StdDev = std(MSE)$; ▷ standard deviation of MSE
    $\beta = min(f(\lambda, X, Y, \alpha))$
    $Fitinfo = AvgMSE, StdDev, \beta, \lambda$; ▷ store values to struct
    $\lambda = \lambda + delta$ ▷ increment lambda until betas are all zero
  $plot(Fitinfo.lambda, Fitinfo.AvgMSE)$
  $BetaIdeal = Fitinfo.Beta(indexMinMSE)$ ▷ select beta vector with min MSE
  $Features = find(BetaIdeal$ =!0) ▷ select features where beta isn't zero
  return $Features$

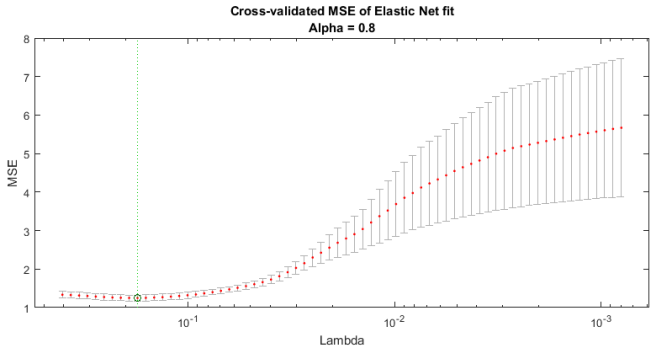

Figure 2. Graphical representation of the Elastic Net pseudocode calculations. The green circle represents the selected λ for this test case (lowest MSE). Therefore, the corresponding coefficients represent the ideal features.

Table I shows the resultant λ values of each test case that minimizes error. Tables XIV to XXII show the results of all the Elastic Net tests and Table III shows the top results of Elastic Net for each test case (see appendix).

Table I
LAMBDA VALUES FOR EACH ELASTIC NET TEST CASE

Test Cases	alpha = 0.8	alpha = 0.2
Healthy vs MCI-1	0.1271	0.5085
Healthy vs MCI-2	0.0825	0.274
Healthy vs AD	0.1112	0.4446
MCI-1 vs MCI-2	0.122	0.3879
MCI-1 vs AD	0.0804	0.2929
MCI-2 vs AD	0.077	0.1008
Healthy vs MCI-1 vs MCI-2	0.1094	0.4802
MCI-1 vs MCI-2 vs AD	0.1217	0.4436
Healthy vs MCI-1 vs MCI-2 vs AD	0.146	0.4417

F. Feature Selection: Information Theory Approach

Different information theoretic algorithms were applied to the clock drawing dataset. These algorithms were Mutual Information (MI) Eq(3), Minimum Redundancy Maximum Relevancy (MRMR) Eq(4), Joint Mutual Information (JMI) Eq(5), and Conditional Mutual Information Maximization (CMIM) Eq(6). This paper used implementations written by Gavin Brown [5].

$$I(X;Y) = \sum_{x \epsilon X} \sum_{y \epsilon Y} p(xy) log(\frac{p(xy)}{p(x)p(y)}) \quad (3)$$

Shannon Mutual Information between two random variables is defined by conditional entropy. Entropy of the class variable, Y, is desired to be very low in order to maximize classification performance. For a given feature X, Mutual Information between X and Y is a measure of the change in entropy of Y due to the presence of X. Therefore, a high Mutual Information between a feature and the class label indicates that the feature is a good predictor of the class label.

For example assume variable Y represented grades in a classroom where half the class was passing and half the class was failing. This would mean Y has high entropy because there's a 50% chance of choosing a passing/failing student (through random guessing). Now, if a feature X was added that represents the amount of hours each student studies, the mutual info between these variables is very high (assume that kids who passed studied more). Therefore variable X reduces the amount of uncertainty in variable Y (we are more certain who passed based on study time). If feature X was completely independent of Y, then p(xy)=p(x)p(y) and the mutual information between variables would be zero.

$$J_{mrmr} = I(X_n;Y) - \frac{1}{n-1} \sum_{k=1}^{n-1} I(X_n;X_k) \quad (4)$$

Minimum Redundancy Maximum Relevancy includes a consideration of redundancy in addition to maximizing the Mutual Information of a feature and the class label. When selecting multiple features, redundancy is the sum over the currently selected features of their Mutual Information with the new feature considered to be added. The purpose of MRMR

is to eliminate redundant features while still selecting a set of features that enables an accurate prediction of the class label.

$$J_{jmi} = I(X_n; Y) - \frac{1}{n-1} \sum_{k=1}^{n-1} [I(X_n; X_k) - I(X_n; X_k|Y)] \quad (5)$$

Joint Mutual Information is equivalent to the First-Order Utility equation provided by Gavin Brown. [5] The first two terms of this equation are the Shannon Mutual Information of the feature with the class label and the negative term for the feature's redundancy with the currently selected features. Lastly, an additional positive term is included for the conditional redundancy. That is, the redundancy between two features given a class label. This term is an indication of first-order interaction between two features that causes that particular pair of features to be useful in the prediction of the class label.

$$J_{cmim} = I(X_n; Y) - max_k[I(X_n; X_k) - I(X_n; X_k|Y)] \quad (6)$$

Conditional Mutual Information Maximization is similar to JMI; however, CMIM takes a pessimistic approach. Instead of taking the sum of all first order interactions (difference between redundancy and conditional redundancy) CMIM only considers the $X_n; X_k$ first order interaction that outputs the maximum score. For this reason, CMIM selects the X_k based on the interaction with an already selected feature X_n that gives the lowest score J. Therefore, if X_k has high redundancy and low conditional redundancy with only one other feature, that X_k will have a low score regardless of the interactions of X_k with other features.

Algorithm 3 Information Theory Feature Selection (ITFS)

procedure ITFS($features, lables, n$)
  for all $features$ **do**
    $score \leftarrow score.concatenate..$
    $..(InfoTheoryAlg(features(i, labels)))$;
  return $score$
  sort $features$ based on $score$;
  $featuresSelected \leftarrow features[0:n]$;

G. Balancing the Data

Once the relevant features are selected using each of the approaches covered above and before they are trained using a classifier, the data must be balanced. An ideal data set for training has an equal amount of samples from each dataset. When the sample sizes of classes are highly skewed, the accuracy on the majority class examples is overwhelmingly higher than the one achieved on the minority classes [13]. This has a significant affect on the clock drawing data because the largest class (AD class) has 59 patients and the smallest class (MCI1 class) only has 26 patients.

For this reason, The Synthetic Minority Oversampling Technique (SMOTE) was used. This technique adds synthetic data points to the smaller classes thus creating a balanced dataset.The first step in SMOTE is defining the K-nearest-neighbors to search for (usually set to five by default) and the amount of synthetic samples to add. The amount of synthetic samples to add is determined by the majority class. For example, if the majority class is 100 samples and a minority class only has 73 samples, 27 synthetic samples would be added to the minority class so it is balanced with the majority class

Once these parameters are set, the algorithm picks a random instance of a minority class and randomly selects one of its KNN. Synthetic instances are then created through interpolation along the vector-space between the randomly selected instance and its KNN. This is done for several instances of each minority class until all the minority classes are the same size as the majority class [13]. Pseudocode illustrating how SMOTE populates a single minority class can be seen in algorithms 4 and 5. Algorithm 4 is pseudocode of a function that uses SMOTE to populate a minority class given the desired amount of added samples and the k nearest neighbors [13]. Algorithm 5 is a function within the SMOTE function that computes the synthetic samples between each minority class instance and its k nearest neighbors through interpolation along the vector-space between them [13].

Algorithm 4 SMOTE

procedure SMOTE(T, N, k) ▷ Input:#minority class examples, Amount of oversampling, #nearestneighbors; Output:synthetic minority class samples
  if $N < 100$ **then**
    Randomize the T minority class samples
    $T = (N/100) * T$
    $N = 100$
  $N = N/100$ ▷ The amount of SMOTE is assumed to be in integral multiples of 100.
  for i =1 to T **do**
    Compute k nearest neighbors for i, and save the indices in the nnarray
    $POPULATE(N, i, nnarray)$

Keep in mind that the main purpose of the pseudocode in algorithm 4 is to determine how many synthetic samples must be created per minority sample and how many random minority samples to iterate through. For example, if N=100, that means twice as many samples are desired (if minority class size is 50, 50 synthetic samples must be created), which means the code will iterate through every minority class instance and create a synthetic instance between the minority class instance and a KNN. If N=50, that means the size of the class should be increased by just 50% (if minority class size is 50, 25 synthetic samples must be created) and the code will iterate through only half the minority class instances). If N=200, then 3x as many samples are desired and the code will iterate through every minority class instance and create 2 synthetic instances between the minority class instance and a KNN.

Algorithm 5 POPULATE

procedure POPULATE($N, i, nnarray$) ▷ Input: #instances to create, original sample index, array of nearest neighbors; Output:N new synthetic samples in Synthetic array
  while $N \neq 0$ **do**
    $nn = random(1, k)$
    for attr = 1 to numattrs **do** ▷ numattrs= Number of attributes
      dif=Sample[nnarray[nn]][attr]-Sample[i][attr]
      $gap = random(0, 1)$
      Synth[newindex][attr]=Sample[i][attr]+gap*dif
    $N = 100$
    newindex++
    N–

The main purpose of the pseudocode in algorithm 5 is to compute the synthetic samples between each minority class instance and its k nearest neighbors. Lets Assume that within a loop the minority class instance is (1,1) and 2 synthetic instances must be created. The code will first select a random KNN about the point (1,1) (lets say it selects (2,2)). The code then chooses a point a random point between (1,1) and (2,2) and that point is the synthetic sample ((1.1,1.1), (1.85,1.85), etc). If 2 synthetic instances must be created, a KNN is randomly selected again and the interpolation process is repeated.

H. Training of the Classifier

Once the relevant features are selected and the data is balanced using SMOTE, the data is then ready to be trained by the classifier. The classifier that was selected was a feed-forward neural network. The neural network was chosen as the selected classifier because they have been particularly successful in classifying medical data. Additionally, they have a varying number parameters one can tune by changing the number of hidden layers and the number of nodes per hidden layer, allowing the network to be aware of varying degrees of complexity within a data set [15]. Moreover, the neural network attained significantly higher results than when other classifiers were used on the clock drawing data such as random forests and SVMs.

Because the data only consisted of 163 samples, the neural networks were kept relatively small to avoid over-fitting. Three different neural network sizes were used: 1 hidden layer of 50 nodes, 2 hidden layers of 10 nodes, and 2 hidden layers of 20 nodes and 10 nodes. Each of these structures were trained on each of the four information theoretic criterion functions covered above in order to determine the highest performer of each test case (see appendix). Ten-fold cross validation was also implemented during the training of the data thus an average number for the performance metric was obtained along with a confidence interval and standard deviation. Moreover, adam was the optimizer used and early stopping was used to prevent over-fitting. It's also important to note that even though the data was trained after using SMOTE to create artificial samples, the performance metrics obtained only pertain to the correct classifications of real samples.

III. Experiments and Results

A. Information Theoretic Results

Experiments were run as outlined in the above approach. Data was collected for each FS criterion. The selected results were chosen according to a ranking criteria (see equations 3,4,5, and 6). The least complex model that obtained the best performance with a narrow confidence interval with as few features chosen as possible was chosen as the best result in each test case (see figure 3).

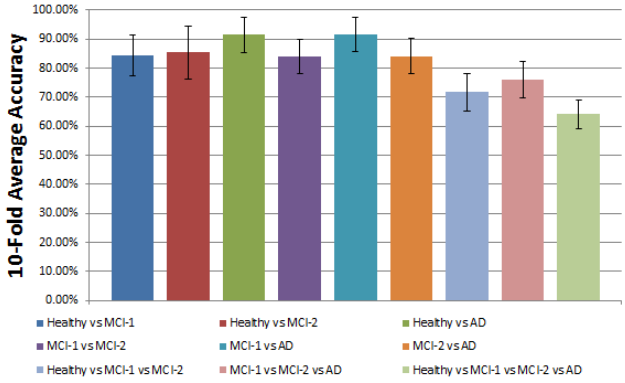

Figure 3. Bar chart of the highest performing results for each test case using information theoretic FS criteria (see table II for values). Error bars correspond to 95% CI.

Based on the information theoretic performance results, each of the binary test cases seemed to significantly outperform the multi-class test cases. All of the binary tests case performances surpassed 80% while the multi-class test cases were all below 80% (see figure 3 and table II). Also by looking at table II it is clear to see that for several of the test cases more than 100 features were used.

B. Elastic Net and Wrapper Results

In addition to the results seen above, these experiments were also replicated on the Elastic Net and wrapper based feature selection algorithms. Elastic Net uses the objective function seen in Eq(1) to select the features based on the mean squared error while also adding in regularization terms for the L1 and L2 norm.

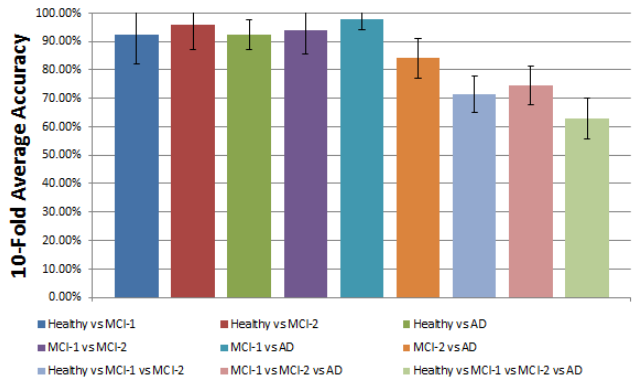

Figure 4. Bar chart of the highest performing results for each test case using Elastic-net technique for feature selection. Error bars correspond to 95% CI.

It was found that Elastic Net achieved on average higher accuracies for the binary experiments while it performed on par with the information theory feature selection in the 3 and 4 class problems (see figure 4). Table III in the appendix shows these results. When using Elastic net, the selected features were all relatively small compared to the information theory results.The 4-class test case attained the highest selected number of features (67 features).

The Recursive Feature Elimination approach performed similarly to the other two approaches for the binary cases, with a notably high performance and small confidence interval for the MCI-1 vs MCI-2 case. This method also achieved the highest accuracies for the 3 and 4 class problems. RFE had the most deviation in its feature selection picking 280 features in the MCI2 vs AD test case but only picking 3,8, and 2 features for healthy vs MCI1, healthy vs MCI2, and MCI1 vs AD respectively. These results are shown in Table IV.

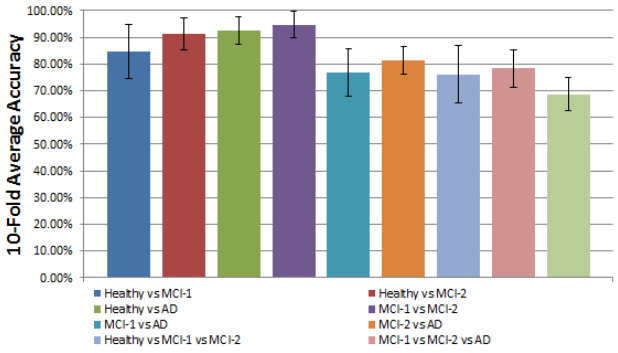

Figure 5. Bar chart of the highest performing results for each test case using RFE. Error bars correspond to 95% CI.

IV. Conclusion

A. Statistical Significance

For the majority of the final results, the confidence intervals of each method overlap, showing that the results are not statistically significantly different from one another. This is not the case for the MCI-1 vs AD result for Elastic Net, which significantly outperformed the other two methods. Within the Information Theoretic approaches (Tables V through XIII) the results are overlapping significantly more. This is due to the similarity of each of the information theoretic formulas, which are all based on mutual information.

It is also worth mentioning that for the most part the binary test cases out performed the multi-class test cases and compared to some of the higher binary results (especially in Elastic Net), the differences are statistically significant. Moreover, the four class test case (healthy vs MCI1 vs MCI 2 vs AD) had the lowest performance in all three feature selection approaches.

B. Applications of Findings

The problem of diagnosing Alzheimer's is one that requires a multidisciplinary set of evaluation techniques in order to acquire a truly accurate diagnosis. However, using only the clock drawing test, the full four-class problem achieved an accuracy of 64%. While this is not useful in and of itself, binary case tests all have accuracies around 80%. The binary test cases will allow us to differentiate between the different classes and provide a preliminary evaluation. It is inexpensive to administer this test, and it does not require any significant amount of training to administer. Since the evaluation is purely objective and consequently does not involve a neuropsychological professional, it can be quickly and regularly administered in a general practitioner's office to provide patients with a preliminary indication of cognitive decline, and the practitioner can recommend the patient for further evaluation by a professional.

Additionally, all of the binary test cases had performances exceeding 80% and for Elastic Net all of the binary results exceeded 90% (excluding MCI2 vs AD). In cases where a professional is unsure whether to diagnose a patient as MCI1 or MCI2, for example, the professional can use the binary classifier to aid in their diagnosis.

C. Future Work

Overall, the binary classifiers performed very well and the three-class and four-class problems did not exceed 80%. A potential solution to the poor performance of the multi-class classification problems would be to implement a one-versus-one ensemble classifier from the binary classifiers. [20] These results can also be further validated and the confidence interval can possibly be reduced with an increased dataset as the dataset will continue to grow.

Additional future work that could be done in attempt to get higher performance is additional tuning of Elastic Net regression in order to derive optimal α values for each test case as well. In the paper only α values of 0.8 and 0.2 were

used; however, an algorithm called LARS-EN can be used to compute Elastic Net regularization paths efficiently [17] to compute better α values.

Another future goal is to shift focus onto the types of features selected from the dCDT and common features selected between FS algorithms. Now that relatively high performance results have been attained, investigation into the most common features that have been selected, whether the selected features are structural or time based in nature, and the development of neuropsychiatric explanations for the selected features.

REFERENCES

[1] Latest Facts & Figures Report — Alzheimer's Association. (2018). Latest Alzheimer's Facts and Figures. [online] Available at: https://www.alz.org/facts/

[2] Díaz-Mardomingo, M., García-Herranz, S., Rodríguez-Fernández, R., Venero, C. and Peraita, H. (2017). Problems in Classifying Mild Cognitive Impairment (MCI): One or Multiple Syndromes?. Brain Sciences, 7(12), p.111.

[3] Umidi, S., Trimarchi, P., Corsi, M., Luzzati, C. and Annoni, G. (2009). CLOCK DRAWING TEST (CDT) IN THE SCREENING OF MILD COGNITIVE IMPAIRMENT (MCI). Archives of Gerontology and Geriatrics, 49, pp.227-229.

[4] Learning classification models of cognitive conditions from subtle behaviors in the digital Clock Drawing Test. (2015). Machine Learning, 102(3), pp.393-441.

[5] G. Brown, "A New Perspective for Information Theoretic Feature Selection," in Proc. of the Twelth International Conference on Artificial Intelligence and Statistics, PMLR 5:49-56, 2009.

[6] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), pp.16-28.

[7] "PEP 8 – Style Guide for Python Code", Python.org, 2018. [Online]. Available: https://www.python.org/dev/peps/pep-0008/. [Accessed: 05- May- 2018].

[8] R. Johnson, "MATLAB Style Guidelines 2.0 - File Exchange - MATLAB Central", Mathworks.com, 2018. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/46056-matlab-style-guidelines-2-0. [Accessed: 05- May- 2018].

[9] V. Bolon-Canedo, B. Remeseiro, A. Alonso-Betanzos, and A. Campilho "Machine Learning for Medical Applications," ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, ISBN 978-287587027-8.

[10] Nezhad, M., Dongxiao Zhu, Xiangrui Li, Kai Yang and Levy, P. (2016). SAFS: A deep feature selection approach for precision medicine. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).

[11] M.Harb, H. and S. Desuky, A. (2014). Feature Selection on Classification of Medical Datasets based on Particle Swarm Optimization. International Journal of Computer Applications, 104(5), pp.14-17.

[12] Santos, V., Datia, N. and Pato, M. (2014). Ensemble Feature Ranking Applied to Medical Data. Procedia Technology, 17, pp.223-230.

[13] Chawla, N., Bowyer, K., Hall, L. and Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal Of Artificial Intelligence Research, 16, pp.321-357.

[14] Beretta, L. and Santaniello, A. (2016). Nearest neighbor imputation algorithms: a critical evaluation. BMC Medical Informatics and Decision Making, 16(S3).

[15] Autio, L., Juhola, M. and Laurikkala, J. (2006). On the neural network classification of medical data and an endeavour to balance non-uniform data sets with artificial data extension. Computers in Biology and Medicine, 37, pp.388-397.

[16] Fonti, V. and Belitser, E. (2017). Feature Selection Using LASSO. Research Paper in Business Analytics.

[17] Zou, H. and Hastie, T. (2003). Regularization and variable selection via the Elastic Net. Statistical Methodology, volume 67, issue 2, pp.301-320.

[18] Davis R, Penney DL, Pittman D, Libon DJ, Swenson R, Kaplan E. The Digital Clock Drawing Test (dCDT) I: Development of a new computerized quantitative system; Montreal, Canada. Paper presented at the The International Neuropsychological Society.2010.

[19] D. Libon, R. Swenson, E. Barnoski and L. Sands, "Clock drawing as an assessment tool for dementia", Archives of Clinical Neuropsychology, vol. 8, no. 5, pp. 405-415, 1993.

[20] Phoenix X. Huang, R. (2018). "Individual feature selection in each One-versus-One classifier improves multi-class SVM performance."

[21] Hidalgo-Muñoz, A., Ramírez, J., Górriz, J. and Padilla, P. (2014). Regions of interest computed by SVM wrapped method for Alzheimer's disease examination from segmented MRI. Frontiers in Aging Neuroscience, 6.

V. Appendix

A. Final Results

Table II
Information Theoretic Final Results

Test Case	FS Criterion	Network Size	Features Selected	Performance	95% CI
SCI vs MCI-1	CMIM	2 L, 20,10 N	25	84.33%	±7.03%
SCI vs MCI-2	CMIM	1 L, 50 N	25	85.42%	±9.11%
SCI vs AD	CMIM	2 L, 20,10 N	125	91.42%	±6.02%
MCI-1 vs MCI-2	MRMR	2 L, 10,10 N	75	84.11%	±5.90%
MCI-1 vs AD	MRMR	2 L, 20,10 N	100	91.49%	±5.99%
MCI-2 vs AD	JMI	2 L, 10,10 N	125	84.05%	±6.14%
SCI vs MCI-1 vs MCI-2	JMI	1 L, 50 N	125	71.64%	±6.46%
MCI-1 vs MCI-2 vs AD	MI	1 L, 50 N	50	75.97%	±6.19%
SCI vs MCI1 vs MCI2 vs AD	MI	1 L, 50 N	100	64.05%	±4.92%

Table III
Elastic Net Final Results

Test Cases	Network Size	Number of Features	Performance	Confidence Interval	alpha
Healthy vs MCI-1	2L, 10,10 N	19	92.29%	±10.24%	0.2
Healthy vs MCI-2	2L, 10,10 N	23	96.03%	±8.84%	0.2
Healthy vs AD	2L, 10,10 N	15	92.31%	±5.36%	0.2
MCI-1 vs MCI-2	1L, 50N	20	93.75%	±8.10%	0.2
MCI-1 vs AD	2L, 20,10 N	28	97.64%	±3.57%	0.2
MCI-2 vs AD	2L, 10,10 N	15	84.05%	±7.01%	0.2
Healthy vs MCI-1 vs MCI-2	2L, 20,10 N	26	71.39%	±6.43%	0.2
MCI-1 vs MCI-2 vs AD	2L, 20,10 N	37	74.41%	±6.85%	0.2
Healthy vs MCI-1 vs MCI-2 vs AD	2L, 10,10 N	67	62.87%	±7.11%	0.2

Table IV
Wrapper Final Results

Test Cases	Network Size	Number of Features	Performance	Confidence Interval
Healthy vs MCI-1	2L, 10,10 N	3	84.67%	±10.15%
Healthy vs MCI-2	2L, 10,10 N	8	91.27%	±5.80%
Healthy vs AD	2L, 10,10 N	58	92.44%	±5.30%
MCI-1 vs MCI-2	1L, 50N	12	94.64%	±4.97%
MCI-1 vs AD	1L, 50N	2	76.90%	±8.80%
MCI-2 vs AD	2L, 20,10 N	280	81.32%	±5.13%
Healthy vs MCI-1 vs MCI-2	1L, 50N	14	76.12%	±10.72%
MCI-1 vs MCI-2 vs AD	1L, 50N	49	78.27%	±7.19%
Healthy vs MCI-1 vs MCI-2 vs AD	1L, 50N	84	68.73%	±6.31%

B. Results From All Test Cases

The tables in this section show the results of each feature selection algorithm using three different neural network hidden layer structures. The number of selected features was also recorded for each value. The least complex model that obtained the best performance with a narrow confidence interval with as few features chosen as possible was chosen as the best result in each test case in the section above (see tables II, III, IV)

Table V
INFO THEORY: SCI VS MCI1

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	80.33% (+/- 10.35%)	74.95% (+/- 10.67%)	76.33% (+/- 9.60%)	82.67% (+/- 16.64%)
	CI + Range	80.33% (+/- 7.80%)	74.95% (+/- 8.05%)	76.33% (+/- 7.24%)	82.67% (+/- 12.55%)
	Number of Features	75	75	125	100
2 Layer, 10,10 Nodes	ACC+STD	77.48% (+/- 8.26%)	78.67% (+/- 12.70%)	79.86% (+/- 16.37%)	84.33% (+/- 9.32%)
	CI + Range	77.48% (+/- 6.23%)	78.67% (+/- 9.58%)	79.86% (+/- 12.34%)	84.33% (+/- 7.03%)
	Number of Features	150	125	75	25
2 Layer, 20,10 Nodes	ACC+STD	78.90% (+/- 10.49%)	74.38% (+/- 13.59%)	80.57% (+/- 15.16%)	82.00% (+/- 13.46%)
	CI + Range	78.90% (+/- 7.91%)	74.38% (+/- 10.24%)	80.57% (+/- 11.43%)	82.00% (+/- 10.15%)
	Number of Features	50	50	125	61

Table VI
INFO THEORY: SCI VS MCI2

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	73.91% (+/- 11.48%)	81.23% (+/- 13.92%)	76.29% (+/- 12.55%)	85.42% (+/- 12.09%)
	CI + Range	73.91% (+/- 8.66%)	81.23% (+/- 10.50%)	76.29% (+/- 9.46%)	85.42% (+/- 9.11%)
	Number of Features	50	75	100	25
2 Layer, 10 Nodes	ACC+STD	69.46% (+/- 9.69%)	75.44% (+/- 12.98%)	80.73% (+/- 8.32%)	80.12% (+/- 13.10%)
	CI + Range	69.46% (+/- 7.30%)	75.44% (+/- 9.78%)	80.73% (+/- 6.28%)	80.12% (+/- 9.88%)
	Number of Features	50	125	75	25
2 Layer, 20,10 Nodes	ACC+STD	70.40% (+/- 8.44%)	77.40% (+/- 11.32%)	78.65% (+/- 6.51%)	83.19% (+/- 12.61%)
	CI + Range	70.40% (+/- 6.37%)	77.40% (+/- 8.53%)	78.65% (+/- 4.91%)	83.19% (+/- 9.51%)
	Number of Features	50	100	100	12

Table VII
INFO THEORY: SCI VS AD

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	88.31% (+/- 5.74%)	90.31% (+/- 10.02%)	89.67% (+/- 8.06%)	90.56% (+/- 7.41%)
	CI + Range	88.31% (+/- 4.33%)	90.31% (+/- 7.56%)	89.67% (+/- 6.08%)	90.56% (+/- 5.59%)
	Number of Features	175	125	125	200
2 Layer, 10 Nodes	ACC+STD	89.44% (+/- 6.71%)	85.78% (+/- 12.74%)	90.42% (+/- 9.53%)	89.33% (+/- 7.05%)
	CI + Range	89.44% (+/- 5.06%)	85.78% (+/- 9.61%)	90.42% (+/- 7.19%)	89.33% (+/- 5.31%)
	Number of Features	200	100	50	150
2 Layer, 20, 10 Nodes	ACC+STD	90.44% (+/- 8.93%)	88.44% (+/- 11.06%)	88.19% (+/- 9.87%)	91.42% (+/- 7.98%)
	CI + Range	90.44% (+/- 6.74%)	88.44% (+/- 8.34%)	88.19% (+/- 7.44%)	91.42% (+/- 6.02%)
	Number of Features	150	125	50	125

Table VIII
INFO THEORY: MCI1 VS MCI2

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	84.88% (+/- 10.79%)	84.05% (+/- 13.06%)	84.23% (+/- 11.43%)	88.15% (+/- 15.36%)
	CI + Range	84.88% (+/- 8.14%)	84.05% (+/- 9.85%)	84.23% (+/- 8.62%)	88.15% (+/- 11.58%)
	Number of Features	100	50	75	50
2 Layer, 10 Nodes	ACC+STD	81.19% (+/- 8.69%)	84.11% (+/- 7.83%)	81.79% (+/- 9.63%)	89.17% (+/- 12.53%)
	CI + Range	81.19% (+/- 6.55%)	84.11% (+/- 5.90%)	81.79% (+/- 7.26%)	89.17% (+/- 9.45%)
	Number of Features	100	75	100	150
2 Layer, 20 Node, 10 Node	ACC+STD	80.71% (+/- 14.59%)	79.29% (+/- 14.87%)	84.52% (+/- 13.43%)	87.56% (+/- 11.09%)
	CI + Range	80.71% (+/- 11.00%)	79.29% (+/- 11.21%)	84.52% (+/- 10.12%)	87.56% (+/- 8.36%)
	Number of Features	50	50	50	200

Table IX
INFO THEORY: MCI1 VS AD

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	89.31% (+/- 11.75%)	90.69% (+/- 10.99%)	90.69% (+/- 10.13%)	91.49% (+/- 7.94%)
	CI + Range	89.31% (+/- 8.86%)	90.69% (+/- 8.29%)	90.69% (+/- 7.64%)	91.49% (+/- 5.99%)
	Number of Features	200	125	100	125
2 Layer, 10 Nodes	ACC+STD	85.97% (+/- 11.36%)	85.65% (+/- 11.88%)	88.61% (+/- 12.45%)	88.29% (+/- 9.99%)
	CI + Range	85.97% (+/- 8.57%)	85.65% (+/- 8.96%)	88.61% (+/- 9.39%)	88.29% (+/- 7.53%)
	Number of Features	75	125	75	75
2 Layer, 20, 10 Nodes	ACC+STD	85.93% (+/- 8.57%)	91.49% (+/- 7.94%)	89.86% (+/- 7.81%)	91.90% (+/- 8.85%)
	CI + Range	85.93% (+/- 6.46%)	91.49% (+/- 5.99%)	89.86% (+/- 5.89%)	91.90% (+/- 6.67%)
	Number of Features	25	100	50	200

Table X
INFO THEORY: MCI2 VS AD

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	79.39% (+/- 8.05%)	82.43% (+/- 11.38%)	81.12% (+/- 8.89%)	84.62% (+/- 10.03%)
	CI + Range	79.39% (+/- 6.07%)	82.43% (+/- 8.58%)	81.12% (+/- 6.71%)	84.62% (+/- 7.56%)
	Number of Features	75	100	100	25
2 Layer, 10 Nodes	ACC+STD	78.62% (+/- 6.85%)	82.14% (+/- 6.33%)	81.41% (+/- 8.80%)	83.91% (+/- 12.85%)
	CI + Range	78.62% (+/- 5.16%)	82.14% (+/- 4.77%)	81.41% (+/- 6.64%)	83.91% (+/- 9.69%)
	Number of Features	25	100	125	25
2 Layer, 20, 10 Nodes	ACC+STD	80.39% (+/- 9.75%)	85.07% (+/- 10.30%)	84.05% (+/- 8.14%)	82.34% (+/- 8.25%)
	CI + Range	80.39% (+/- 7.35%)	85.07% (+/- 7.76%)	84.05% (+/- 6.14%)	82.34% (+/- 6.22%)
	Number of Features	150	100	125	12

Table XI
INFO THEORY: SCI VS MCI1 VS MCI2

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	67.66% (+/- 8.59%)	69.75% (+/- 10.30%)	71.64% (+/- 8.56%)	66.50% (+/- 13.17%)
	CI + Range	67.66% (+/- 6.48%)	69.75% (+/- 7.77%)	71.64% (+/- 6.46%)	66.50% (+/- 9.93%)
	Number of Features	75	75	125	50
2 Layer, 10 Nodes	ACC+STD	68.03% (+/- 9.77%)	64.37% (+/- 12.45%)	62.43% (+/- 10.07%)	69.05% (+/- 15.43%)
	CI + Range	68.03% (+/- 7.36%)	64.37% (+/- 9.39%)	62.43% (+/- 7.59%)	69.05% (+/- 11.64%)
	Number of Features	75	75	100	50
2 Layer, 20, 10 Nodes	ACC+STD	65.71% (+/- 10.79%)	66.15% (+/- 11.87%)	67.76% (+/- 14.48%)	67.56% (+/- 12.16%)
	CI + Range	65.71% (+/- 8.13%)	66.15% (+/- 8.95%)	67.76% (+/- 10.92%)	67.56% (+/- 9.17%)
	Number of Features	75	50	75	100

Table XII
INFO THEORY: MCI1 VS MCI2 VS AD

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	75.97% (+/- 8.21%)	73.00% (+/- 12.51%)	73.83% (+/- 8.78%)	72.06% (+/- 9.14%)
	CI + Range	75.97% (+/- 6.19%)	73.00% (+/- 9.43%)	73.83% (+/- 6.62%)	72.06% (+/- 6.89%)
	Number of Features	50	125	125	275
2 Layer, 10 Nodes	ACC+STD	74.14% (+/- 8.40%)	72.06% (+/- 11.29%)	69.06% (+/- 9.62%)	73.32% (+/- 8.77%)
	CI + Range	74.14% (+/- 6.33%)	72.06% (+/- 8.51%)	69.06% (+/- 7.26%)	73.32% (+/- 6.62%)
	Number of Features	100	100	100	225
2 Layer, 20, 10 Nodes	ACC+STD	71.69% (+/- 8.93%)	71.51% (+/- 13.31%)	67.16% (+/- 6.62%)	72.35% (+/- 13.00%)
	CI + Range	71.69% (+/- 6.74%)	71.51% (+/- 10.04%)	67.16% (+/- 4.99%)	72.35% (+/- 9.81%)
	Number of Features	100	125	75	100

Table XIII
INFO THEORY: SCI VS MCI1 VS MCI2 VS AD

Network Parameters	DATA	MI	MRMR	JMI	CMIM
1 Layer, 50 Nodes	ACC+STD	64.05% (+/- 6.52%)	64.71% (+/- 7.70%)	64.25% (+/- 13.44%)	65.69% (+/- 10.14%)
	CI + Range	64.05% (+/- 4.92%)	64.71% (+/- 5.81%)	64.25% (+/- 10.14%)	65.69% (+/- 7.64%)
	Number of Features	100	75	75	225
2 Layer, 10 Nodes	ACC+STD	65.08% (+/- 10.32%)	63.32% (+/- 11.58%)	62.38% (+/- 8.26%)	66.33% (+/- 9.96%)
	CI + Range	65.08% (+/- 7.78%)	63.32% (+/- 8.73%)	62.38% (+/- 6.23%)	66.33% (+/- 7.51%)
	Number of Features	150	125	75	200
2 Layer, 20, 10 Nodes	ACC+STD	64.27% (+/- 10.36%)	63.98% (+/- 12.05%)	61.84% (+/- 12.66%)	64.26% (+/- 12.57%)
	CI + Range	64.27% (+/- 7.81%)	63.98% (+/- 9.09%)	61.84% (+/- 9.55%)	64.26% (+/- 9.48%)
	Number of Features	125	100	50	200

Table XIV
WRAPPER+ELASTIC NET: SCI VS MCI1

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	79.76% (+/- 7.84%)	90.95% (+/- 12.32%)	88.29% (+/- 13.61%)
	CI+Range	79.76% (+/- 5.91%)	90.95% (+/- 9.29%)	88.29% (+/- 10.26%)
	Number of Features	3	17	19
2 Layer, 10 Nodes	ACC+STD	84.67% (+/- 13.46%)	86.95% (+/- 13.51%)	92.29% (+/- 13.59%)
	CI+Range	84.67% (+/- 10.15%)	86.95% (+/- 10.19%)	92.29% (+/- 10.24%)
	Number of Features	3	17	19
2 Layer, 20,10 Nodes	ACC+STD	74.90% (+/- 20.17%)	88.33% (+/- 15.45%)	86.62% (+/- 11.42%)
	CI+Range	74.90% (+/- 15.21%)	88.33% (+/- 11.65%)	86.62% (+/- 8.61%)
	Number of Features	3	17	19

Table XV
WRAPPER+ELASTIC NET: SCI VS MCI2

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	90.67% (+/- 12.29%)	90.85% (+/- 10.43%)	91.61% (+/- 13.03%)
	CI+Range	90.67% (+/- 9.27%)	90.85% (+/- 7.87%)	91.61% (+/- 9.82%)
	Number of Features	8	23	23
2 Layer, 10 Nodes	ACC+STD	91.27% (+/- 7.69%)	93.67% (+/- 6.41%)	96.03% (+/- 8.84%)
	CI+Range	91.27% (+/- 5.80%)	93.67% (+/- 4.83%)	96.03% (+/- 6.67%)
	Number of Features	8	23	23
2 Layer, 20,10 Nodes	ACC+STD	87.66% (+/- 7.16%)	92.06% (+/- 10.43%)	91.90% (+/- 11.30%)
	CI+Range	87.66% (+/- 5.40%)	92.06% (+/- 7.87%)	91.90% (+/- 8.52%)
	Number of Features	8	23	23

Table XVI
WRAPPER+ELASTIC NET: SCI VS AD

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	89.08% (+/- 8.36%)	93.89% (+/- 6.71%)	86.44% (+/- 9.09%)
	CI+Range	89.08% (+/- 6.30%)	93.89% (+/- 5.06%)	86.44% (+/- 6.86%)
	Number of Features	58	13	15
2 Layer, 10 Nodes	ACC+STD	92.44% (+/- 7.02%)	90.44% (+/- 10.22%)	92.31% (+/- 7.11%)
	CI+Range	92.44% (+/- 5.30%)	90.44% (+/- 7.71%)	92.31% (+/- 5.36%)
	Number of Features	58	13	15
2 Layer, 20,10 Nodes	ACC+STD	91.56% (+/- 8.84%)	88.56% (+/- 11.94%)	91.64% (+/- 8.84%)
	CI+Range	91.56% (+/- 6.66%)	88.56% (+/- 11.94%)	91.64% (+/- 6.66%)
	Number of Features	58	13	15

Table XVII
WRAPPER+ELASTIC NET: MCI1 VS MCI2

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	94.64% (+/- 6.59%)	84.17% (+/- 17.66%)	93.75% (+/- 10.74%)
	CI+Range	94.64% (+/- 4.97%)	84.17% (+/- 13.32%)	93.75% (+/- 8.10%)
	Number of Features	12	18	20
2 Layer, 10 Nodes	ACC+STD	91.73% (+/- 8.85%)	93.75% (+/- 11.52%)	91.31% (+/- 7.22%)
	CI+Range	91.73% (+/- 6.67%)	93.75% (+/- 8.69%)	91.31% (+/- 5.44%)
	Number of Features	12	18	20
2 Layer, 20,10 Nodes	ACC+STD	91.31% (+/- 9.64%)	88.39% (+/- 11.10%)	91.13% (+/- 9.72%)
	CI+Range	91.31% (+/- 7.27%)	88.39% (+/- 8.37%)	91.13% (+/- 7.33%)
	Number of Features	12	18	20

Table XVIII
WRAPPER+ELASTIC NET: MCI1 VS AD

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	76.90% (+/- 11.68%)	88.99% (+/- 10.07%)	92.60% (+/- 9.65%)
	CI+Range	76.90% (+/- 8.80%)	88.99% (+/- 7.60%)	92.60% (+/- 7.28%)
	Number of Features	2	16	28
2 Layer, 10 Nodes	ACC+STD	74.54% (+/- 15.19%)	88.33% (+/- 9.01%)	91.90% (+/- 10.15%)
	CI+Range	74.54% (+/- 11.45%)	88.33% (+/- 6.79%)	91.90% (+/- 7.65%)
	Number of Features	2	16	28
2 Layer, 20,10 Nodes	ACC+STD	68.77% (+/- 14.47%)	91.81% (+/- 11.59%)	97.64% (+/- 4.73%)
	CI+Range	68.77% (+/- 10.91%)	91.81% (+/- 8.74%)	97.64% (+/- 3.57%)
	Number of Features	2	16	28

Table XIX
WRAPPER+ELASTIC NET: MCI2 VS AD

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	79.23% (+/- 10.61%)	96.00% (+/- 6.63%)	83.14% (+/- 8.05%)
	CI+Range	79.23% (+/- 8.00%)	96.00% (+/- 5.00%)	83.14% (+/- 6.07%)
	Number of Features	280	56	15
2 Layer, 10 Nodes	ACC+STD	78.03% (+/- 9.38%)	97.27% (+/- 5.82%)	84.05% (+/- 9.29%)
	CI+Range	78.03% (+/- 7.07%)	97.27% (+/- 4.39%)	84.05% (+/- 7.01%)
	Number of Features	280	56	15
2 Layer, 20,10 Nodes	ACC+STD	81.32% (+/- 6.80%)	96.07% (+/- 4.84%)	82.82% (+/- 11.29%)
	CI+Range	81.32% (+/- 5.13%)	96.07% (+/- 3.65%)	82.82% (+/- 8.51%)
	Number of Features	280	56	15

Table XX
WRAPPER+ELASTIC NET: SCI VS MCI1 VS MCI2

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	76.12% (+/- 14.22%)	64.60% (+/- 18.84%)	71.65% (+/- 16.75%)
	CI+Range	76.12% (+/- 10.72%)	64.60% (+/- 14.20%)	71.65% (+/- 12.63%)
	Number of Features	14	11	26
2 Layer, 10 Nodes	ACC+STD	70.99% (+/- 12.61%)	63.01% (+/- 13.11%)	64.46% (+/- 12.05%)
	CI+Range	70.99% (+/- 9.51%)	63.01% (+/- 9.88%)	64.46% (+/- 9.09%)
	Number of Features	14	11	26
2 Layer, 20,10 Nodes	ACC+STD	75.03% (+/- 19.85%)	60.43% (+/- 8.64%)	71.39% (+/- 8.53%)
	CI+Range	75.03% (+/- 14.96%)	60.43% (+/- 6.52%)	71.39% (+/- 6.43%)
	Number of Features	14	11	26

Table XXI
WRAPPER+ELASTIC NET: MCI1 VS MCI2 VS AD

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	78.27% (+/- 9.53%)	65.63% (+/- 15.57%)	71.46% (+/- 11.16%)
	CI+Range	78.27% (+/- 7.19%)	65.63% (+/- 11.74%)	71.46% (+/- 8.42%)
	Number of Features	49	26	37
2 Layer, 10 Nodes	ACC+STD	75.26% (+/- 12.32%)	73.64% (+/- 9.37%)	66.72% (+/- 16.82%)
	CI+Range	75.26% (+/- 9.29%)	73.64% (+/- 7.06%)	66.72% (+/- 12.68%)
	Number of Features	49	26	37
2 Layer, 20,10 Nodes	ACC+STD	78.11% (+/- 13.94%)	69.45% (+/- 12.76%)	74.41% (+/- 9.08%)
	CI+Range	78.11% (+/- 10.51%)	69.45% (+/- 9.62%)	74.41% (+/- 6.85%)
	Number of Features	49	26	37

Table XXII
WRAPPER+ELASTIC NET: SCI VS MCI1 VS MCI2 VS AD

Network Parameters	DATA	Wrapper	Elastic Net (a=0.8)	Elastic Net (a=0.2)
1 Layer, 50 Nodes	ACC+STD	68.73% (+/- 8.37%)	51.27% (+/- 8.05%)	65.39% (+/- 11.40%)
	CI+Range	68.73% (+/- 6.31%)	51.27% (+/- 6.07%)	65.39% (+/- 8.60%)
	Number of Features	84	25	67
2 Layer, 10 Nodes	ACC+STD	66.51% (+/- 10.82%)	55.95% (+/- 14.34%)	62.87% (+/- 9.43%)
	CI+Range	66.51% (+/- 8.16%)	55.95% (+/- 10.81%)	62.87% (+/- 7.11%)
	Number of Features	84	25	67
2 Layer, 20,10 Nodes	ACC+STD	62.31% (+/- 11.54%)	53.18% (+/- 14.90%)	69.22% (+/- 16.18%)
	CI+Range	62.31% (+/- 8.70%)	53.18% (+/- 11.23%)	69.22% (+/- 12.20%)
	Number of Features	84	25	67