Early Diagnosis of Alzheimer's Disease Using Machine Learning On Cognitive Tests

May 15th 2019

Timothy Duong, Nick Klein, Kyle Naddeo,
Thai Nghiem, Lonnie L. Souder II

***Abstract*— The clock drawing test is an noninvasive, early diagnostic tool being used to assess a patients level of cognitive impairment and help practitioners diagnose Alzheimer's disease. Traditionally, the way a neuropsychologist performs a clock drawing test is by asking patients to draw an analog clock showing the time 10 past 11 on a piece of paper. This is a subjective and time costly process, with a significant rate of misdiagnosis. Now, with the advance in technology, researchers were able extract hundreds of features from the clock drawing test using a smart pen. These features are uniquely suited for machine learning algorithms to expose the relationship between the feature space and the degree of cognitive impairment. In this research, three feature selection techniques are used to extract most important features for a diagnosis: a genetic algorithm, a wrapper feature selection method, and a hierarchical architecture feature selection algorithm. The selected features are then used to train two different classifiers: a neural network and a stacked generalization. These classifiers are used to analyze the features as to diagnose each patient as SCI (Subtle Cognitive Impairment), MCI (Mild Cognitive Impairment), or AD (Alzheimer's Disease). Using Stacked Generalization to combine the different feature selection techniques into one network increased the performance in most test cases. The results indicate a differential diagnosis accuracy of high 70% to mid 80% are achievable. These results can potentially allow general health practitioners to expedite the process of using the clock drawing test as a preliminary screening tool for Alzheimer's Disease and other cognitive impairment diagnosis .**

I. Introduction

A. Alzheimer's Disease and Mild Cognitive Impairment Background

There is major requisite in the health care community to accurately diagnose Alzheimer's disease (AD); it is estimated that someone is diagnosed every 65 seconds in the US [1] . The current state of diagnosis in primary care is based on the subjective suspicions of the care givers [2]. This method can lead to delayed, missed or misdiagnoses of the disease. Although there is no proven cure for AD, early detection can allow for medication regimens that can delay the diseases progress and allow for more end of life care planning. Unfortunately, misdiagnosing can lead to health effects from unneeded medication and additional stress on patient and family; it is estimated that 2 in 10 AD cases may be misdiagnosed [3].

A common misdiagnosis occurs when a patient has Mild Cognitive Impairment (MCI) which also involves memory problems and noticeable changes in a patients functionality. Approximately 15 to 20% of people the age of 65 or older have MCI and it is a known prodrome of AD [4]. Due to this MCI has gained much more attention and from the research there has been different categories identified: amnestic (aMCI), dysexecutive (dMCI) and a mixed phenotype (mxMCI).

B. Clock Drawing Test

The Clock Drawing Test (CDT) is a neuropsychiatric assessment to screen for AD and MCI developed in the 1960's. The assessment became popular in 1983 when it was added to the Boston Diagnostic Aphasia Examination [5]. The test is broken into two components: auditory and visual. The first is to draw an analog clock after the patient has been told the time. This is to evaluate the patients comprehension to auditory commands. The second test is a visuospatial test where the patient is asked to simply copy a given clock. In this part of the test the clock is not shown during the replication process and therefore the patient must rely on memory.

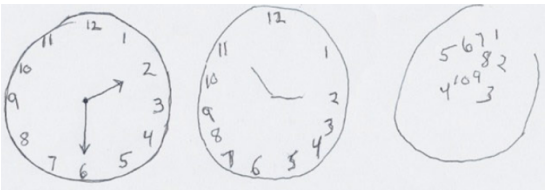

Fig. 1: Example of the the auditory component of the Clock Drawing Test [6].
Left: A healthy patient with full comprehension of the test; it is crucial to see that the clock is still wrong (the hour hand should be between the 2 and 3) even though the patient is healthy.
Middle: Attempt of a patient with MCI where the spacing of the numbers is nonuniform but the basic features of the clock still exist.
Right: The attempt of a patient with AD; sadly there is reminisce of comprehension (there are numbers) but the basic idea of a clock is lost.

The most important detail in Figure 1 is the healthy patients test. It shows the subtleties that are associated with the CDT; not every incorrect clock is a sign of dementia. To correctly interpret the CDT factors beyond just the final image are needed. In 2015 the digital CDT or dCDT was established by Digital Cognition Technologies after 10 years of research and development at Lahey Hospital Medical Center and MIT [7]. This modern version of the CDT allows for these unseen factors to be archived for future researchers where as before they were only seen by the administer of the test. These factors include such things as timeliness of completion of each number or the order in which the numbers were written. The dCDT will generate 351 such factors for every test conducted. The method used to gather these factors is not through the use of a tablet which may alter the reaction of the patients; rather, a digital pen was developed to allow the patients to write on paper.

C. Machine Learning

Machine learning (ML) is a subset of Artificial Intelligence which is based in the interpretation of large data sets. The true goal of ML is to perform a task that was not explicitly programmed; instead, the machine must identify patterns underlying the data set. If a supervised ML algorithm is successful it should be able to analyze a training data set (a data set with features and labels); learn the patterns then accurately label unseen feature combinations.

The concept of the curse of dimensionality illustrates that the increased number of possibilities due to the increased number of features is nonlinear and a single extra feature can vastly complicate the models task. Conversely, the more features available gives a more well defined task. With an infinite amount of training instances the number of features has no relevance (aside from computational cost); even if 90% of the features are irrelevant, in theory the model will eventually learn this fact. Unfortunately, the dCDT data set is in its infancy; only containing 163 instances. This means that the ML model will not be able to analyze the subtleties of all the features to accurately predict. The remedy is feature selection in pursuit of finding the least amount of features to still accurately represent the task. An inherent byproduct of feature selection will be the suggestions of why certain features are more informative to diagnosis's. This may give medical professionals a mathematical grounding in their subjective beliefs or redirect them to new insights. The classification problems tested in these experiments were SCI vs MCI1, SCI vs MCI2, SCI vs AD, MCI1 vs MCI2, MCI1 vs AD, MCI2 vs AD, SCI vs MCI1 vs MCI2, MCI1 vs MCI2 vs AD, and SCI vs MCI1 vs MCI2 vs AD.The types of feature selection methods used in this study will be discussed in the Approach section.

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 [8]
- MATLAB Programming Style Guidelines [15]

2) Constraints - This study was constrained by the following

- Size of the data set (163 instances)
- Model Parameters - avoidance of over or under fitting
- Vectorization of dCDT features which may be incomplete

II. Approach

A. Feature Selection

1) Genetic Algorithm

Genetic algorithms are a subset of iterative optimization algorithms inspired by Darwin's Theory of Evolution. In essence, the solution to the problem is encoded into a genome on which the program can more easily operate and evaluate. The nature of this encoding is application specific, thus it can take many forms such as trees, arrays, or bit fields. Then, through a rough simulation of nature, the genome is coerced towards its optimal value or set of values.

There are many variations of genetic algorithms, but all of them use the same basic principles. First, of course, the possible solutions must be encoded into something on which the algorithm can easily operate. This mimics DNA in nature as it is the sole contributor to the characteristics of the individual solution. In our algorithm, since we are performing feature selection, our genome consists of a vector of booleans where True represents an active feature and False represents an inactive features. This is referred to as the *genome* or *individual*. Also, all genetic algorithms are iterative as to explore many possible solutions. All of the possibilities explored in one iteration are referred to collectively as a *generation*. A set of randomly initialized individuals must be created as a starting point for the algorithm. This is generation 0. Henceforth, during each iteration, the following events will take place. Each individual must be evaluated to determine its fitness as a solution. For this application, the genomes were used to choose active features to train a simple multinomial logistic regression classifier. The test accuracy of said classifier was used as the fitness score for each individual. Next, the algorithm must in some way simulate natural selection, whether this be through a tournament style selection or simply taking the top individuals in terms of fitness. In our application, we only take the top 30 individuals to survive to the next generation. Next, the algorithm will perform some form of mating or crossover between the remaining survivors. The hope is that by crossing over the best individuals, eventually, you may arrive at a *super individual* which displays the best features from its parents. The final step of any genetic algorithm is mutation. In the mutation step, there is an individual probability that each individual may have some random change to one or many of its features. There is a lot of room for experimentation here in terms of the probability that an individual will mutate, the probability that each feature will mutate, and how much and in what way a feature will mutate. For our algorithm all individuals experience a very small mutation in the form of a gene toggle for 1% of the features.

2) Hierarchical Feature Selection

Hierarchical feature selection methods involve the use of two or more traditional feature selection methods in cascade, refining the number and importance of features as they are evaluated by each of the traditional feature selection methods. During these experiments the traditional methods used are Information Theory Feature Selection, a filter method, and Wrapper Methods. The order is determined so that the more computational expensive method is cascaded behind the less computational expensive method. In this case, Wrapper Methods is cascaded after Information Theory Feature Selection. Each of these methods have sub-methods listed and explained below.

a) Information Theory

Information Theory Feature Selection scores features based on the Mutual Information (MI) shared by the label and a given feature which ranks the relevancy of features. Several advanced algorithms are also used which expand on the mutual information algorithm and take into account dependency between the feature and label as well as independence between features. The advanced Information Theory techniques used are Minimum Redundancy Maximum Relevancy (MRMR), Joint Mutual Information (JMI), and Conditional Mutual Information Maximization (CMIM). This paper used implementations written by Gavin Brown [9] to be consistant with previous research performed by Binaco et al [10]. Once the features have been ranked, a threshold is determined which determines the cutoff score below which features are removed from the relevant feature set used by the classifier [11].

- Mutual Information:

$$I(X;Y) = \sum_{x \epsilon X} \sum_{y \epsilon Y} p(xy) log(\frac{p(xy)}{p(x)p(y)}) \quad (1)$$

Mutual Information is based on work by Shannon [12] and expanded upon by Brown to create principled methodology to expand Shannon's work to apply to mutual information between features and a class label. Brown's retrofit of Shannon Mutual Information states the mutual information between random variables are defined by conditional entropy. Entropy of class labels are desired to be low which maximises performance of the classifier. Mutual information of a feature, X, and the feature label, Y, is a measure of how much entropy is observed in Y due to the presence of X. An example is when Y is a label for the contents of a beverage glass. Half the glasses contain cola while the other half contain brewed tea. Y has high entropy since there is a 50% chance of choosing either beverage with no other knowledge. A feature X, representing the color of the beverage would have low mutual information as both beverages are brown. Another feature X, representing carbonation would have high mutual information which reduces the uncertainty in the label Y since cola is carbonated and tea is not. In this instance when feature X is independent of Y, then $p(xy) = p(x)p(y)$ and the mutual information between the feature and label is zero.

- Minimum Redundancy Maximum Relevancy:

$$J_{mrmr} = I(X_n;Y) - \frac{1}{n-1} \sum_{k=1}^{n-1} I(X_n;X_k) \quad (2)$$

Minimum Redundancy Maximum Relevancy considers when features selected using Mutual Information are redundant in addition to the MI process. These redundant features are removed to further parse down a feature set while still selecting the best possible set of features for an accurate prediction of class label. For multiple features, redundancy is the sum of the Mutual Information with the new feature over the currently selected features.

- Joint Mutual Information:

$$J_{jmi} = I(X_n;Y) - \frac{1}{n-1} \sum_{k=1}^{n-1} [I(X_n;X_k) - I(X_n;X_k|Y)] \quad (3)$$

Joint Mutual Information is equivalent to the First-Order Utility equation provided by Gavin Brown [9]. The JMI technique is equivalent to MRMR with an added term for conditional redundancy relating the label to the new and current features selected. This term indicates the usefulness of a feature pair used to predict a label as opposed to the usefulness of each label alone.

- Conditional Mutual Information Maximization:

$$J_{cmim} = I(X_n;Y) - max_k[I(X_n;X_k) - I(X_n;X_k|Y)] \quad (4)$$

Conditional Mutual Information Maximization is the most recent criterion used, developed by Fluret (2004)[9]. It is similar to JMI but it examines the information between a feature and the target label compared with each feature. This pessimistic approach takes in only the difference between the redundancy of two features and their conditional redundancy that outputs a maximum score. Based on this, the new features selected for removal from the set are those that give the lowest score with high redundancy and low conditional redundancy with another feature. These conditions show the selected feature will have a low score in all cases.

b) Wrapper

Wrapper methods are greedy search algorithms that find the optimal features by running each combination of through the classifier and develop weights for features based on their inclusion in successful classifications. For this, the classifier becomes a black box and the outcome of the classification becomes the objective function. These exhaustive searches become exponentially more computationally more expensive with an increase in the number of features[11].

- Recursive Feature Elimination (RFE):
Recursive Feature Elimination with a Linear SVM Kernel (SVM-RFE) is a sequential searching method that reduces a complete set of features by evaluating the current feature set and removing the feature with the lowest weight in the resultant weight vector of the linear SVM. The process is performed until a single feature remains. The reverse order of removal becomes the ranked order of the feature set. The feature set which resulted in the highest SVM classification is used as the optimal set of features[13].

- Sequential Feature Algorithms (SFA):
Sequential Feature Algorithms are a family of wrapper methods which function similar to RFE except that instead of using weights to eliminate features recursively, SFA(s) remove and add features based on a user defined classifier performance metric designed to reduce the feature set to a set number of features, k. The SFS family has four members, these are Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), Sequential Forward Floating Selection (SFFS), and Sequential Backward Floating Selection (SBFS). The forward algorithms begin with an empty feature set and begin by including a single feature which generates the highest accuracy. Each feature is chosen in this method until a number of features, k are chosen. Backward algorithms have their features chosen by beginning with the complete feature set and removing a single feature in similar fashion until it is left with a feature set of k members. The floating variants are extensions of the former algorithms which have an extra inclusion/exclusion step that adds removed features or removes currently selected features to check for instances in which a feature was included or excluded due to the order in which it was checked, to the detriment of the classification accuracy. If this step results in a feature set with a number of features not equal to k, the first step is repeated.

B. Classification

Before training the classifiers, the Synthetic Minority Oversampling Technique (SMOTE) was used in order to balance the data, since the sample size of the classes are skewed (the largest class has 59 patients and the smallest class only

has 26 patients). SMOTE added synthetic data points to the smaller classes, thus created a balanced dataset that have 200 instances based on the original 163.

1) Neural Network

After important features are found and the data is balanced using SMOTE, we used it to train the feed-forward neural network classifier. The neural network was chosen because it has varying number parameters one can tune for the network to be aware of varying degrees of complexity within a data set [14]. From experiments, neural network is also found to attain significantly higher results than when other classifiers were used on the clock drawing data such as random forests and SVMs.

To avoid over-fitting due to the data limited data, which only consisted of 163 samples, the neural networks were kept relatively shallow. 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. Ten-fold cross validation, adam optimizer, and early stopping were also used to prevent over-fitting. Notably, although we were using SMOTE to create artificial samples, the performance metrics obtained only pertain to the correct classifications of real samples.

2) Stacked Generalization

Stacked Generalization is a technique in machine learning to minimize the error of a single classifier by taking advantage of the different error in multiple classifiers. A new meta classifier is trained on the outputs of two or more tier 1 classifiers to learn how best to combine the outputs of the tier 1 classifiers to improve overall accuracy.

In the original data set, there are 163 data points and after balancing the data set through SMOTE there are approximately 200 data points. For this implementation of Stacked Generalization, the SMOTED data set was randomly divided into six different subsets. One subset is reserved to test the overall network and will be called the Stacked Generalization Test subset. The remaining five subset will be used to train and test the tier 1 classifiers. In this implementation of Stacked Generalization, six different tier 1 classifiers were used. Each tier 1 classifier used a different subset of features that were dependant on different feature selection methods.

In one fold of training, the tier 1 classifiers are trained on 4 of the 5 subsets of the data. The tier 1 classifiers then make predictions based on the remaining subset of data. The output from the tier 1 classifiers from the remaining subset of data is used to train the meta classifier. The output from the tier 1 classifiers are saved to train the meta classifier after all folds of training are complete. For example, in the four class problem, there are four nodes in the tier 1 classifiers on the output layer. The meta classifier trains and evaluates based on the output of this layer. In the next fold of training, the same tier 1 classifiers are trained and evaluated again. However, this time a different group of 4 subsets are used for training and a different subset is used to test the tier 1 classifiers. The output from the tier 1 classifiers in this fold are saved again. This process repeats until all the subsets of data have been used for testing.

In order to evaluate the Stacked Generalization, one final pass through the tier 1 classifiers needs to be performed. Through this final pass, all 5 subsets of the data are used to train the tier 1 classifiers. After the tier 1 classifiers are trained, they are evaluated using the Stacked Generalization Test subset. The SMOTED data samples are removed from this dataset. The output from this pass is used to evaluate the meta classifier.

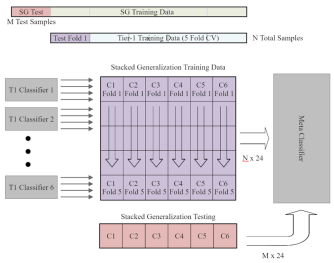

Fig. 2: Stacked Generalization Architecture

III. Experiments and Results

A. Genetic Algorithm

Number of Features	25	50	100	351
Accuracy [%]	57	69	63	57

TABLE I: Results of feature selection through genetic algorithms for a 4-class problem using multinomial logistic regression as the test classifier

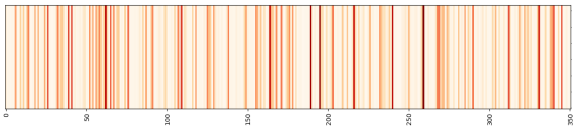

Fig. 3: Representation of how often features were chosen by the genetic algorithm in the top 50 surviving individuals; Lighter shades correspond to fewer occurrences, while darker shades of red correspond to multiple occurrences of an active gene.

Test Cases	FS Criterion	Max Features	Performance
Healthy vs MCI-1	GA	50	95.2%
Healthy vs MCI-2	GA	50	100%
Healthy vs AD	GA	50	80.4%
MCI-1 vs MCI-2	GA	50	100%
MCI-1 vs AD	GA	50	92.3%
MCI-2 vs AD	GA	50	100%
Healthy vs MCI-1 vs MCI-2	GA	50	81.2%
MCI-1 vs MCI-2 vs AD	GA	50	89.7%
Healthy vs MCI-1 vs MCI-2 vs AD	GA	50	61.1%

TABLE II: Performance of logistic regression using features selected by the genetic algorithm for multiple classification problems; This is significant because the genomes in the genetic algorithm were evaluated based on the performance of a multinomial logistic function fit to the features

Test Cases	FS Criterion	Neural Network Size	Max Features	Performance	95% Confidence Interval
Healthy vs MCI-1	GA	2 Layer, 20-10 Nodes	50	59.8%	8.7%
Healthy vs MCI-2	GA	2 Layer, 20-10 Nodes	50	59.1%	6.5%
Healthy vs AD	GA	2 Layer, 20-10 Nodes	50	69.5%	7.9%
MCI-1 vs MCI-2	GA	2 Layer, 20-10 Nodes	50	66.2%	6.5%
MCI-1 vs AD	GA	2 Layer, 20-10 Nodes	50	58.5%	8%
MCI-2 vs AD	GA	2 Layer, 20-10 Nodes	50	60.3%	12.5%
Healthy vs MCI-1 vs MCI-2	GA	2 Layer, 20-10 Nodes	50	51.9%	8.5%
MCI-1 vs MCI-2 vs AD	GA	2 Layer, 20-10 Nodes	50	63.7%	12.3%
Healthy vs MCI-1 vs MCI-2 vs AD	GA	2 Layer, 20-10 Nodes	50	68.3%	8.7%

TABLE III: Performance of a neural network using features selected by the genetic algorithm for multiple classification problems

B. Hierarchical Feature Selection

Using the Gavin Brown approaches to Information Theory Feature Selection used by the SPPRL in their 2018 paper [10],

each classification problem had features selected by each of the four types of Information Theory with tests using 75, 100, and 125 as the maximum allowable number of features output. The output of each test was used as the input for both the RFE and SFS Wrappers. The features output by the second phase were classified using three separate neural networks; a single layer feed forward perceptron model with 50 neurons, a double layer FF perceptron model with 20 and 10 neurons respectively and, a double layer FF perceptron model with 10 and 10 neurons. The best accuracies with corresponding confidence intervals are shown in Figure 4 for tests using the RFE wrapper with a corresponding table of best results in Figure 5.

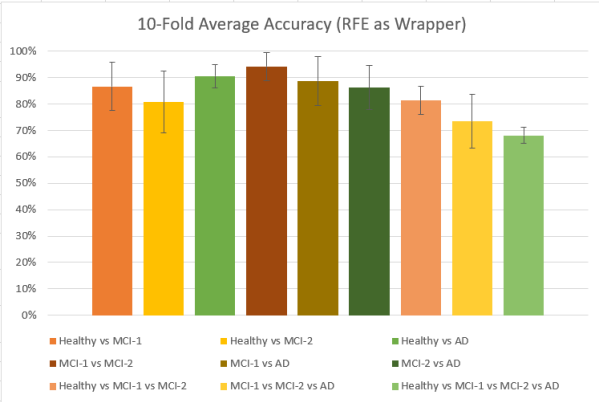

Fig. 4: Bar chart of the highest performing results for each test case using information theoretic FS criteria refined by step forward FS. Error bars correspond to 95% CI.

Test cases	FS Criterion	Neural Network Size	Features Selected	Performance	95% Confidence Interval
Healthy vs MCI-1	JMI → RFE	1 Layer, 50 Nodes	9	86.71%	9.06%
Healthy vs MCI-2	MI → RFE	2 Layer, 10-10 Nodes	8	80.73%	11.71%
Healthy vs AD	MI → RFE	2 Layer, 20-10 Nodes	24	90.56%	4.45%
MCI-1 vs MCI-2	MRMR → RFE	2 Layer, 20-10 Nodes	16	94.23%	5.38%
MCI-1 vs AD	CMIM → RFE	1 Layer, 50 Nodes	33	88.75%	9.18%
MCI-2 vs AD	JMI → RFE	2 Layer, 20-10 Nodes	58	86.16%	8.33%
Healthy vs MCI-1 vs MCI-2	JMI → RFE	2 Layer, 20-10 Nodes	18	81.37%	5.41%
MCI-1 vs MCI-2 vs AD	CMIM → RFE	2 Layer, 20-10 Nodes	49	73.49%	10.19%
Healthy vs MCI-1 vs MCI-2 vs AD	MI → RFE	1 Layer, 50 Nodes	73	68.16%	3.01%

Fig. 5: Best results for each classification using Recursive Feature Selection and Information Theory.

A bar chart showing the best accuracies with corresponding confidence intervals for tests using the SFS wrapper is shown by Figure 6 with a corresponding table of best results in Figure 7. The combined best results of all tests are represented by Figure 8. All two class problems were found to be classifiable with accuracy in the high 80s or low 90s while the drop off for the three and four class problems became very pronounced. Classification involving the AD class dropped to the mid 70% range for three class problems and down to 69% for the four class problem.

When compared against the results from tests using Information theory alone, Hierarchical Feature Selection often proved valuable in reducing the number of necessary features to determine an equally certain accuracy of similar magnitude. Only in a few circumstances were accuracies improved for a classification problem. The tests resulting in significantly improved accuracy are MCI1 vs MCI2 improved from 84.11% to 94.21%, SCI vs MCI1 vs MCI2 improved from 71.64% to 81.37%, and the four class problem which improved from 64.05% to 68.16%. Complete lists of

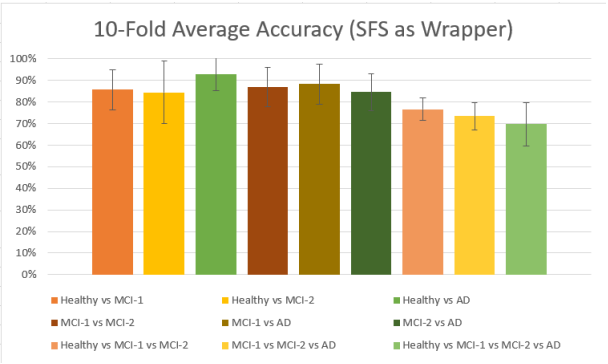

Fig. 6: Bar chart of the highest performing results for each test case using information theoretic FS criteria refined by recursive features elimination FS. Error bars correspond to 95% CI.

Test cases	FS Criterion	Neural Network Size	Features Selected	Performance	95% Confidence Interval
Healthy vs MCI-1	MRMR → SFS	1 Layer, 50 Nodes	15	85.76%	9.23%
Healthy vs MCI-2	CMIM → SFS	1 Layer, 50 Nodes	30	84.52%	14.60%
Healthy vs AD	CMIM → SFS	1 Layer, 50 Nodes	32	92.89%	7.61%
MCI-1 vs MCI-2	MI → SFS	1 Layer, 50 Nodes	72	86.96%	9.21%
MCI-1 vs AD	MRMR → SFS	2 Layer, 20-10 Nodes	90	88.33%	9.04%
MCI-2 vs AD	JMI→ SFS	2 Layer, 20-10 Nodes	73	84.55%	8.59%
Healthy vs MCI-1 vs MCI-2	CMIM → SFS	2 Layer, 20-10 Nodes	85	76.70%	5.32%
MCI-1 vs MCI-2 vs AD	CMIM → SFS	1 Layer, 50 Nodes	90	73.52%	6.36%
Healthy vs MCI-1 vs MCI-2 vs AD	MI → SFS	1 Layer, 50 Nodes	118	69.67%	9.99%

Fig. 7: Best results for each classification using Step Forward Feature Selection and Information Theory

Test cases	FS Criterion	Neural Network Size	Features Selected	Performance	95% Confidence Interval
Healthy vs MCI-1	JMI → RFE	1 Layer, 50 Nodes	9	86.71%	9.06%
Healthy vs MCI-2	CMIM → SFS	1 Layer, 50 Nodes	30	84.52%	14.60%
Healthy vs AD	CMIM → SFS	1 Layer, 50 Nodes	32	92.89%	7.61%
MCI-1 vs MCI-2	MRMR → RFE	2 Layer, 20-10 Nodes	16	94.23%	5.38%
MCI-1 vs AD	CMIM → RFE	1 Layer, 50 Nodes	33	88.75%	9.18%
MCI-2 vs AD	JMI → RFE	2 Layer, 20-10 Nodes	58	86.16%	8.33%
Healthy vs MCI-1 vs MCI-2	MRMR → RFE	2 Layer, 20-10 Nodes	18	81.37%	5.41%
MCI-1 vs MCI-2 vs AD	CMIM → SFS	1 Layer, 50 Nodes	90	73.52%	6.36%
Healthy vs MCI-1 vs MCI-2 vs AD	MI → SFS	1 Layer, 50 Nodes	118	69.67%	9.99%

Fig. 8: Best results for each classification using Hierarchical Feature Selection.

comparisons of the best results from Information Theory and Hierarchical feature selection for each of the two class problems are displayed in Figures 9 and 10. A similar comparison of results for three and four class problems are displayed in Figure 11.

Problem Cases / Feature Selection Method	SCI vs MCI1 (2 classes)			SCI vs MCI2 (2 classes)			SCI vs AD (2 classes)		
	Accuracy	*Confidence Interval*	*Feature Used*	*Accuracy*	*Confidence Interval*	*Feature Used*	*Accuracy*	*Confidence Interval*	*Feature Used*
Information Theory **only**	84.33%	± 7.03%	25	85.42%	± 9.11%	25	91.42%	± 6.02%	125
Information Theory **refined** by Wrapper	86.71%	± 9.06%	9	84.52%	± 14.6%	30	92.89%	± 7.61%	32

Fig. 9: A comparison of the best results from Information Theory alone and Information Theory with added Wrapper using using two classes - Part 1.

C. Stacked Generalization

For all test runs of the different class combinations for Stacked Generalization used the scheme mentioned in the Stacked Generalization description. All the the different class problems used the same feature selection methods: Reduce

Problem Cases / Feature Selection Method	MCI1 vs MCI 2 (2 classes)			MCI1 vs AD (2 classes)			MCI2 vs AD (2 classes)		
	Accuracy	*Confidence Interval*	*Feature Used*	*Accuracy*	*Confidence Interval*	*Feature Used*	*Accuracy*	*Confidence Interval*	*Feature Used*
Information Theory **only**	84.11%	± 5.90%	75	91.49%	± 5.99%	100	84.05%	± 6.14%	125
Information Theory **refined** by Wrapper	94.23%	± 5.38%	16	88.75%	±9.18%	33	86.16%	± 8.33%	58

Fig. 10: A comparison of the best results from Information Theory alone and Information Theory with added Wrapper using using two classes - Part 2.

Problem Cases / Feature Selection Method	SCI vs MCI1 vs MCI2 (3 classes)			MCI1 vs MCI2 vs AD (3 classes)			SCI vs MCI1 vs MCI3 vs AD (4 classes)		
	Accuracy	*Confidence Interval*	*Feature Used*	*Accuracy*	*Confidence Interval*	*Feature Used*	*Accuracy*	*Confidence Interval*	*Feature Used*
Information Theory **only**	71.64%	± 6.46%	125	75.97%	± 6.19%	50	64.05%	± 4.92%	100
Information Theory **refined** by Wrapper	81.37%	± 5.41%	18	73.52%	± 6.36%	90	68.16%	± 3.01%	73

Fig. 11: A comparison of the best results from Information Theory alone and Information Theory with added Wrapper using using three and four classes.

Feature Elimination, Linear Support Vector Classifier, and Mutual Information. The only problem that differed was the four class problem which used the three previously mention feature selection in addition to Genetic Algorithm, Information Theory with Wrapper and the previous years selected features. The results can be found in figure 12.

Test Cases	FS Chosen	Neural Network T1 Classifier	Performance	95% Confidence
Healthy vs MCI-1	RFE, MI, LSVC	1 layer, 10 nodes	83.62%	9.12%
Healthy vs MCI-2	RFE, MI, LSVC	1 layer, 10 nodes	87.14%	8.17%
Healthy vs AD	RFE, MI, LSVC	1 layer, 10 nodes	95.00%	4.32%
MCI-1 vs MCI-2	RFE, MI, LSVC	1 layer, 10 nodes	89.14%	7.23%
MCI-2 vs AD	RFE, MI, LSVC	1 layer, 10 nodes	87.62%	8.52%
Healthy vs MCI-1 vs MCI-2	RFE, MI, LSVC	1 layer, 10 nodes	78.28%	7.43%
MCI-1 vs MCI-2 vs AD	RFE, MI, LSVC	1 layer, 10 nodes	83.68%	8.03%
Healthy vs MCI-1 vs MCI-2 vs AD	RFE, MI, LSVC, GA, IT	1 layer, 10 nodes	80.55%	6.53%

Fig. 12: Stacked Generalization Scores-5 Fold

Comparing the results from figure 12 with those from figure 7 and figure 5, there is significant improvement in the accuracy of the network for higher class problems. In the four class problem, there is an improvement of approximately 10% using the Stacked Generalization Architecture. An interesting observation during test runs of the Stacked Generalization were the variation in the accuracy of the tier 1 classifiers during the testing stage of the network. Some of the tier 1 classifiers scored nearly 100% accuracy during some folds. This issue is probably due to the small original data set. In the testing set, there are approximately 30 samples with SMOTE. However, when this testing set was selected, the samples were randomly chosen from a SMOTED data set. Therefore, in certain iterations of the test, there may be fewer than 20 real samples the Stacked Generalization was evaluated on. The easiest way to fix this issue is to get more data samples for the original data set. Another approach is to change the number of splits so that the Stacked Generalization test set is larger. However, this causes the training set to be reduced which may impact performance.

D. Most Frequently Picked Features

From the three lists of features picked by three separate FS techniques (genetic algorithm, wrapper method, and hierarchical FS algorithm), we manually chose the features that are picked by either 2 or 3 FS techniques and compare their results to the ones picked by only 1 FS technique. The classifier we used to evaluate the performance was the neural network discussed in Section II-B.1, and the test case was the 4-class problem (Healthy, MCI-1, MCI-2, and AD). The results of this experiment can be seen in Figure 13.

Used Features Picked by	Accuracy	Confidence Interval	Feature Used	Neural Network Size
1 FS algorithm	70.69%	± 9.65%	118	1 Layer, 50 Nodes
2 FS algorithms	62.97%	± 7.17%	54	2 Layers, 10-10 Nodes
3 FS algorithms	46.95%	± 11.24%	5	1 Layer, 50 Nodes

Fig. 13: Results of the experiment for the 4-class problem

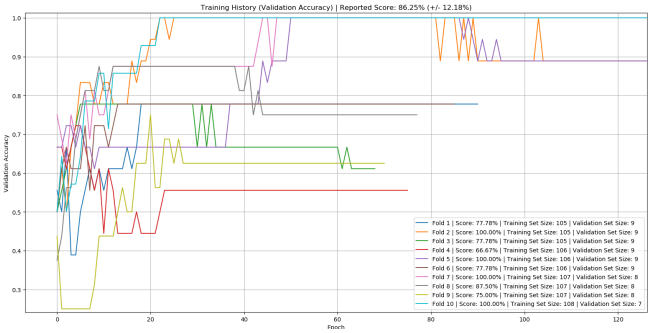

Fig. 14: Initial replication study results over 10 folds. Reported score: $86.25\%(\pm 12.18\%)$.

From the figure, it is evident that the classier used features selected by more FS techniques had lower accuracy than the one used features selected by fewer FS techniques. For this reason, no further effort should be given into investigating this method.

E. Replication Study

A replication study was performed on the CDT data in an effort to validate the results found in [10]. The study attempted to replicate the highest reported performance, i.e. 91.49% $(\pm 5.99\%)$ accuracy on the MCI-1 vs. AD dataset using a feedforward MLP with two hidden layers (having 20 nodes and 10 nodes, respectively). The dataset was augmented with synthetic data samples generated using the SMOTE algorithm. In addition, these results were obtained using the top 100 features as selected using Minimum Redundancy Maximum Relevancy (MRMR) feature selection.

The results reported in [10] were obtained by performing a stratified 10-fold cross-validation on the MCI-1 vs. AD dataset, training a neural network on each fold, and averaging the highest validation accuracies obtained for each fold during training.

An initial attempt to replicate the conditions of the experiment using the experiment's original code yielded the training histories seen in Figure 14.

While the minor discrepancy between the reported and observed results is attributable to the random initialization of the network weights, several things are immediately evident upon viewing Figure 14. Firstly, there is an enormous variance in the performance of each individual network: the lowest-performing network settles at 55.56% accuracy (though it reports 66.67% accuracy as it achieved this performance briefly in the first few training epochs), while the highest-performing networks achieve 100% accuracy. In addition, the size of the validation dataset is extraordinarily small; with a sample size of only 9, a single instance's correct or incorrect classification may affect the performance by over 10%.

To evaluate the effect of validation set size on network performance, the same experiment was performed, this time over 5-fold cross-validation (Figure 15).

Compared to the previous experiment, the performance of the networks sharply decreases. This may be attributable to

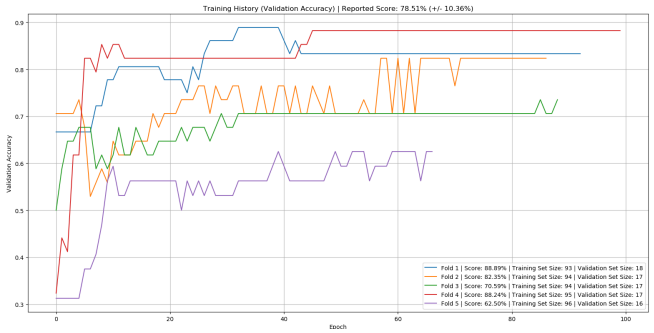

Fig. 15: Evaluating neural network performance using 5-fold cross-validation. Reported score: $78.51\%(\pm 10.26\%)$.

the fact that while the size of the validation set doubled ($9 \rightarrow 18$), the size of the training dataset shrank ($105 \rightarrow 93$), and so the networks were deprived of information that would have maintained their performance.

To verify the effect of a highly skewed testing/training split, an experiment was designed using the same dataset with the same features, using an identical neural network architecture (this time, written in original code). Instead of 10-fold cross-validation, ten runs of training were performed with a 90:10 ratio of training and testing data, with the testing data drawn uniformly from the total dataset (and ensured to be non-synthetic examples). This experiment confirmed that a high ratio of training data to testing data led to an increased performance, reporting an average of 90% accuracy over ten runs.

From these experiments, we conclude that while the results reported in [10] are valid, they are potentially misleading as they are reported on a very small validation dataset with a very high variance between individual runs. More study is required to further validate these results, using much larger datasets with validation splits having a larger absolute number of samples.

IV. Conclusions

A. Statistical Significance

Three instances of truely improved performance in the classifier are observed when compared to the results of the SPPRL's findings in *Automated Analysis of the Clock Drawing Test for Differential Diagnosis of Mild Cognitive Impairment and Alzheimers Disease*. The hierarchial methods improved the MCI-1 vs MCI-2 accuracy from 84.11% to 94.23% and the SCI vs MCI-1 vs MCI-2 from 71.64% to 81.37%. The Stacked Generalization method increased performance in most test cases when compared to when individual neural networks were trained on a single subset of features. The best results in the case of the four class problem result from the Stacked Generalization method which achieved a performance of 80.55% which is approximately 10% higher than what the other individual classifiers that used the feature selection methods mentioned in this study.

B. Applications of Findings

When trying to diagnose Alzheimer's Disease, an automated method should have a 90% accuracy minimally to determine the risk a patient has for developing worse symptoms of the disease. In the best case, the methods here give a 80.55% accuracy for the four class problem. However, all binary and triary classifications have at least one method that provides accuracies in the high 80% and low 90% which can be used to refine a diagnosis where the probability of one or more labels is unlikely. In all circumstances these results give doctors some concrete data and a place to begin their own investigation especially since applying this version of the clock drawing test requires little training to implement in a general practitioner's office before recommending a patient see a specialist.

C. Future Work

These experiments suffered from a lack of data. If the research could continue with thousands of patients to be studied, it's quite likely that a method that could classify a patient's cognitive state with high accuracy above 90% could be developed. Further research of hierarchical methods could include three or more feature selection algorithms in cascade or to include new feature selection methods such as the genetic algorithm developed for this case. A further analysis of Stacked Generalization can be performed to see how different architectures affect the overall performance. For example, one can increase the number of tier 1 classifiers or change the number of layers in each tier 1 classifier.
Additionally the feature selection methods were able to find 54 features which were selected by two of the three feature selection algorithms (Wrapper, Hierarchical, and Genetic) and 5 features selected by all of them. This allows further research to focus on a smaller subset of features and see which other features can supplement them best or which may be parsed out when more data is acquired.
Finally a classifier which rules out individual classes can be made from the best algorithm of each classification problem where results can be refined to give a more definitive diagnosis.

V. Bibliography

References

[1] Alzheimer's Disease and Dementia. (2019). Facts and Figures. [online] Available at: https://www.alz.org/alzheimers-dementia/facts-figures [Accessed 2 May 2019].

[2] Bradford, A., Kunick, M., Schulz, P., Williams, S. and Singh, H. (2009). Missed and Delayed Diagnosis of Dementia in Primary Care: Prevalence and Contributing Factors. [online] NCBI. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2787842/R12 [Accessed 13 May 2019].

[3] Reinberg, S. (2019). 2 in 10 Alzheimer's Cases May Be Misdiagnosed. [online] WebMD. Available at: https://www.webmd.com/alzheimers/news/20160726/2-in-10-alzheimers-cases-may-be-misdiagnosed1 [Accessed 13 May 2019].

[4] Alzheimer's Disease and Dementia. (2019). Mild Cognitive Impairment (MCI). [online] Available at: https://www.alz.org/alzheimers-dementia/what-is-dementia/related$_conditions/mild - cognitive - impairment[Accessed13May2019]$.

[5] Eknoyan, D., Hurley, R. and Taber, K. (2012). The Clock Drawing Task: Common Errors and Functional Neuroanatomy. The Journal of Neuropsychiatry and Clinical Neurosciences, 24(3), pp.260-265.

[6] Burnett, D. (2019). Why Is Clock-Drawing Used in Cognitive Tests, Like the One Trump Took?. [online] The Cut. Available at: https://www.thecut.com/2018/01/clock-drawing-and-trumps-cognitive-test.html [Accessed 13 May 2019].

[7] Digitalcogtech.com. (2019). Digital Cognition Technologies — About Us. [online] Available at: http://www.digitalcogtech.com/about [Accessed 13 May 2019].

[8] Python.org. (2019). PEP 8 – Style Guide for Python Code. [online] Available at: https://www.python.org/dev/peps/pep-0008/ [Accessed 13 May 2019].

[9] 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.

[10] R Binaco, N Calzaretto, J Epifano, S McGuire, M Umer, R Polikar, *"Automated Analysis of the Clock Drawing Test for Differential Diagnosis of Mild Cognitive Impairment and Alzheimers Disease"* 2018

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

[12] Shannon, C. (1948). A Mathematical theory of communication, Bell syst. *Tech. J.*, 27(3):379-423

[13] Hidalgo-Muoz, A., Ramrez, J., Grriz, J. and Padilla, P. (2014). Regions of interest computed by SVM wrapped method for Alzheimers disease examination from segmented MRI. Frontiers in Aging Neuroscience, 6.

[14] 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.

[15] Datatool.com. (2019). [online] Available at: http://www.datatool.com/downloads/MatlabStyle2%20book.pdf [Accessed 13 May 2019].