ANN based Data Mining Analysis of the Parkinson's Disease

Satish M. Srinivasan
School of Graduate Professional Studies
Penn State University
Malvern, PA 19355

Michael Martin
Penn State University
University Park, PA 16802

Abhishek Tripathi
The College of New Jersey
Ewing, NJ 08628

ABSTRACT
This paper intends to provide an evaluation of the different pre-processing techniques that can aid a classifier in the classification of the Parkinson Disease (PD) dataset. PD is a chronic and progressive moment disorder caused due to the malfunction and death of vital nerve cells in the brain. The key indications of the chronic malady in the central nervous system can be best captivated from the Mentation, Activities of Daily Life (ADL), Motor Examination, and Complications of Therapy. The speech symptom which is an ADL is a common ground for the progress of the PD. A comprehensive study on the application of different pre-processing techniques is carried out on the PD dataset obtained from the UCI website. For classifying the PD dataset we employed the ANN based MLP classifier. With the objective of improving the prediction accuracies of the healthy, and people with Parkinson disease on the PD dataset this study highlights the fact that the combination of several pre-processing techniques namely Discretization, Resampling, and SMOTE can best aid in the classification process. This study is unique in the sense that we have not come across any similar studies in the Data Mining literature.

General Terms
Data Mining, Pre-processing techniques, classification techniques.

Keywords
ANN, MLP, Discretization, Resampling, SMOTE, Classification

1. INTRODUCTION
Parkinson's disease (PD) is a progressive disorder of the nervous system that affects movement. Symptoms related to this disorder develop gradually, continue, and worsen over the period of time. The cause of this disease is rarely known and there are over a millions of people round the world who are suffering from PD. A very limited number of options are available to cure and manage the symptoms of this disease [1]. The symptoms differ in different stages of the disease but in general they all involve cognitive and behavioral problems [2]. Some of the common signs and symptoms related to PD are Tremor (Shaking of limb, hand, fingers), slowed movement, rigid muscles, impaired posture and balance, loss of automatic movements, speech changes, writing changes etc. [3]. The death of the vital nerve cells also known as neurons in the area of the brain called the substantia nigra paves the way for the PD. The neurons are responsible for producing a chemical substance commonly known as dopamine which sends messages to the part of the brain that controls movement and coordination. As PD progresses the amount of dopamine produced in the brain decreases, leaving a person unable to control the movement normally [1].

In this paper we provide a comprehensive data mining analysis on the PD dataset obtained from the UCI, using the Artificial Neural Network (ANN) classifier. More specifically, this analysis seeks to demonstrate how the different pre-processing techniques affect the results of classification on the PD dataset using an ANN based algorithm. The specific algorithm used in this study is the Multi-Layer Perceptron (MLP) function. In the next section we will provide a brief discussion about the ANN based MLP classifier and the PD dataset used in this study.

2. ANN BASED CLASSIFIER AND PD DATASET
An ANN is a computational structure inspired by the study of the biological neural processing. There are several different types of ANN ranging from simple to very complex. ANN represents a highly parallelized dynamic system with a directed graph topology that can receive the output information by means of a reaction of its state on the input actions. The ANN node provides a variety of feed forward networks that are commonly called as back-propagation networks. The back-propagation refers to the method for computing the error gradient for a feed forward network. A general topology of the ANN consists of several input neurons that receive a normalized numerical input from each of the variables in the dataset. These normalized values are then multiplied by factors, known as connection weights. The net product of these multiplications are summed up and become the net inputs that enter into an activation function that calculates the outputs of the hidden neurons [4]. ANN based MLP classifier is a conventional three-layer MLP network. This network consists an input layer, a hidden layer, and an output layer. Each layer consists of a several number of input nodes or neurons. These numbers are usually determined by the attributes in the dataset that is being used. For each attribute in the dataset a neuron is included in the input layer. The number of neurons in the hidden layer is usually determined using a trial and error method and the output layer consists of a number of neurons that represents a rage, demonstrating the disease classification. The MLP function uses a back propagating method to classify instances [4].

The PD dataset used in this paper is taken from the UCI machine learning repository [5]. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 of which were digonized with Parkinson's disease. Each column in the table is a particular voice measure, and each row corresponds to one of the 195 voice records from the individuals. The objective of this dataset is to discriminate

healthy people from those with PD which are represented as 0 and 1 respectively. Across the 23 columns and 195 rows the data is represented as a numeric value. In this dataset 48 instances were classified as 0's (healthy) and 147 instances were classified as 1's (with PD). Our initial assertion about the PD dataset is that it is unbalanced. The table 1 below describes the attributes in the PD dataset [4].

Table 1. PD dataset description [1, 4, 6]

Attribute	Description
MDVP: Fo (Hz)	Average vocal fundamental frequency
MDVP: Fhi (Hz)	Maximum vocal fundamental frequency
MDVP: Flo (Hz)	Minimum vocal fundamental frequency
MDVP: Jitter (%)	Measure of variation in fundamental frequency
MDVP: Jitter (Abs)	Measure of variation in fundamental frequency
MDVP: RAP	Measure of variation in fundamental frequency
MDVP: PPQ	Measure of variation in fundamental frequency
Jitter: DDP	Measure of variation in fundamental frequency
MDVP: Shimmer	Measure of variation in fundamental frequency
MDVP: Shimmer (dB)	Measure of variation in fundamental frequency
Shimmer: APQ3	Measure of variation in fundamental frequency
Shimmer: APQ5	Measure of variation in fundamental frequency
MDVP: APQ	Measure of variation in fundamental frequency
Shimmer: DDA	Measure of variation in fundamental frequency
RPDE	Nonlinear dynamical complexity measures
D2	Nonlinear dynamical complexity measures
DFA	Signal fractal scaling exponent
Spread 1	Nonlinear measure of fundamental frequency variation
Spread 2	Nonlinear measure of fundamental frequency variation
PPE	Nonlinear measure of fundamental frequency variation
NHR	Measure of ratio of noise to tonal components in the voice
HNR	Measure of ratio of noise to tonal components in the voice
Concept Class	Healthy, Sick

3. METHOD

The PD dataset was obtained from [5] as a CSV file and was converted to an ARFF file. This ARFF file was then loaded in to WEKA for further analysis. In this dataset we examined for missing values and duplicate records. Across each attribute, it was confirmed that there are no missing values in the dataset. Additionally, using WEKA's *RemoveDuplicates* function, the data was checked for any duplicated values/records but there were none. Once the data was confirmed to be clean from duplicates and missing values, we proceeded to our next step *i.e.* pre-processing and adjusting the parameters of the algorithms. The following are the different pre-processing techniques considered in this study:

1. **Attribute Selection**: This pre-processing step seeks to remove any irrelevant attributes from the dataset. This step can be performed using the WEKA's *AttributeSelection* function and by evaluating the attributes based off the predictability. The settings used in this study can be summarized as (*WEKA/Filter/Supervised/Attribute/AttributeSelection {False, False}*)
2. **Discretize**: This pre-processing step converts all the numeric values to nominal values. This step can be performed using the WEKA's *Discretize* function. The settings used in this study can be summarized as (*WEKA/Filter/Supervised/Attribute/Discretize {First-Last, binRangePrecision: 6})*.
3. **Resample with 100% of Data:** This pre-processing step produces a subsample of the data set. One can define whether to use sampling with or without replacement. This step was performed using WEKA's *Resample* function to sample 100% of the data without replacement. The settings used in this study can be summarized as *(WEKA/Filter/Supervised/Instance/Resample {1,100%})*.
4. **SMOTE Method:** This pre-processing step resamples the data set using the *Synthetic Minority Oversampling Technique* (SMOTE), that uses the k-Nearest Neighbour (KNN) method. This step results in a much more balanced dataset. The settings used in this study can be summarized as *(WEKA/Filter/Supervised/Instance/SMOTE {0,5,100%,1})*.
5. **Attribute Selection and Discretization:** In this pre-processing step the dataset was first filtered to remove any irrelevant attributes and then the dataset was discretized. The parameter settings for each of these steps are identical to those listed above.
6. **Attribute Selection and Resample @ 100%**: In this pre-processing step the dataset was first filtered to remove any irrelevant attributes and then the dataset was resampled. The parameter settings for each of these steps are identical to those listed above.
7. **Attribute Selection and SMOTE Method**: In this pre-processing step the dataset was first filtered to remove any irrelevant attributes and then the SMOTE was applied. The parameter settings for each of these steps are identical to those listed above.
8. **Discretize and Resample @ 100%:** In this pre-processing step the dataset was first discretized and then resampled. The parameter settings for each of these steps are identical to those listed above.
9. **Discretize and SMOTE Method:** In this pre-processing step the dataset was first discretized and then the SMOTE method was applied. The parameter settings for each of these steps are identical to those listed above.
10. **Resample and SMOTE Method:** In this pre-processing step the dataset was first resampled and then the SMOTE method was applied. The parameter settings for each of these steps are identical to those listed above.
11. **Attribute Selection, Discretization and SMOTE Method:** In this pre-processing step the dataset was

first filtered to remove any irrelevant attributes, then discretized, and finally the SMOTE method was applied to balance the dataset. The parameter settings for each of these steps are identical to those listed above.

12. **Resample, SMOTE and Discretize:** In this pre-processing step the dataset was first resampled at 100%. The SMOTE method was then applied on this dataset. Finally, the resultant dataset was discretized. The parameter settings for each of these steps are identical to those listed above.

All these pre-processing steps were performed on the PD dataset using the version 3.8 of WEKA. Only those pre-processing step that involved the SMOTE method was conducted in the version 3.6 of WEKA. To perform cross-validation, three different approaches were considered namely the 10-fold cross validation, 80:20 split and 70:30 split. Across each pre-processing step, for each of the cross-validation approaches a confusion matrix was obtained. Using the confusion matrix 14 different performance metrics were computed. The table 2 below describes the notations used for each of the pre-processing step and the list of the performance metrics computed in this study.

Table 2. List of Performance measures and the notations for the pre-processing steps

Abbreviation	Description
K	Kappa Statistics
I	Bookmaker Informedness
ROC	ROC Area
TPR	Sensitivity
TNR	Specificity
PPV	Precision
NPV	Negative Predictive Value
FPR	False Positive Rate
FDR	False Discovery Rate
FNR	Miss Rate
ACC	Accuracy
F1	F1-score
MCC	Matthews Correlation Coefficient
MK	Markedness
AS	Attribute Selection
D	Discretize
R	Resample (100%)
SM	SMOTE
AS+D	Attribute Selection + Discretize
AS+R	Attribute Selection + Resample
AS+SM	Attribute Selection + SMOTE
D+R	Discretize + Resample (100%)
D+SM	Discretize + SMOTE
R+SM	Resample (100%) + SMOTE
AS+D+SM	Attribute Selection + Discretize + SMOTE
R+SM+D	Resample + SMOTE + Discretize

The table 3 summarizes the derivation strategy for all the performance metrics. In the next section we will briefly describe the experiments and the results obtained.

Table 3. Metrics for evaluating the performance of the classifier

True Positive Rate (TPR): $TPR = \frac{TP}{P} = \frac{TP}{TP+FN}$

True Negative Rate (TNR): $TNR = \frac{TN}{N} = \frac{TN}{TN+FP}$

Positive Predictive Value (PPV): $PPV = \frac{TP}{TP+FP}$

Negative Predictive Value (NPV): $NPV = \frac{TN}{TN+FN}$

False Negative Rate (FNR): $FNR = \frac{FN}{P} = \frac{FN}{TP+FN} = 1 - TPR$

False Discovery Rate (FDR): $FDR = \frac{FP}{TP+FP} = 1 - PPV$

False Positive Rate (FPR): $FPR = \frac{FP}{N} = \frac{FP}{TN+FP} = 1 - TNR$

False Omission Rate (FOR): $FOR = \frac{FN}{TN+FN} = 1 - NPV$

Accuracy (ACC): $ACC = \frac{TP+TN}{P+N}$

F_1-score: $F1 = 2 * \frac{PPV*TPR}{PPV+TPR} = \frac{2TP}{2TP+FP+FN}$

Matthews Correlation Coefficient (MCC):

$$(TP * TN - FP * FN)/\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$$

Bookmaker Informedness (I): $I = TPR + TNR - 1$

Markedness (MK): $MK = PPV + NPV - 1$

4. EXPERIMENT, RESULT AND DISCUSSION

Several experiments were performed using WEKA to classify the PD dataset. For all the experiments the ANN based MLP classifier was employed as the classification algorithm. Each time a different pre-processing step (see table 2) was performed on the PD dataset. After pre-processing, the dataset was portioned in to training and testing datasets. For cross-validation three different approaches namely 10-fold, 80:20 split, and 70:30 split were performed. The 10-fold cross-validation was performed on the entire dataset. In case of the 80:20 and 70:30 split the training dataset comprises of 80% and 70% of the instances respectively, and the test dataset comprises of the remaining 20% and 30% of the instances respectively.

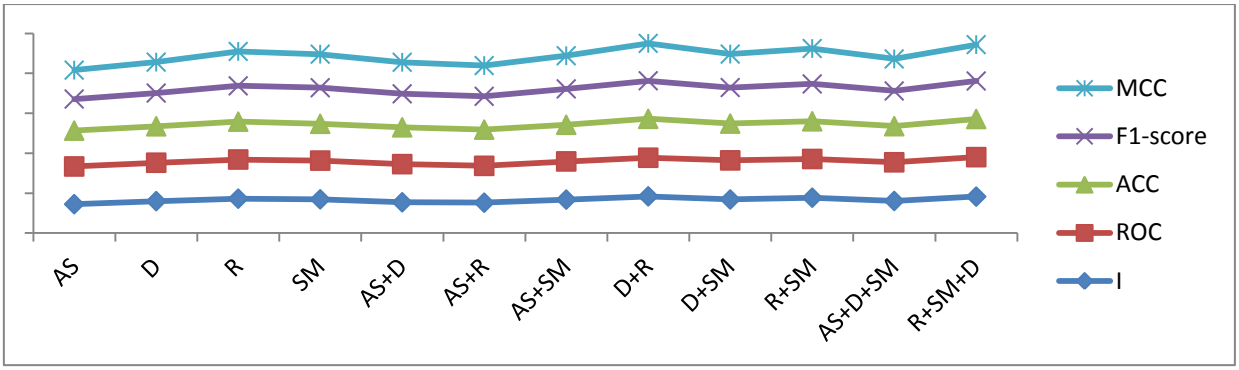

Figure 1. 10-fold cross validation performance measures (MCC, F1-score, ACC, ROC and I) across 12 different pre-processing steps

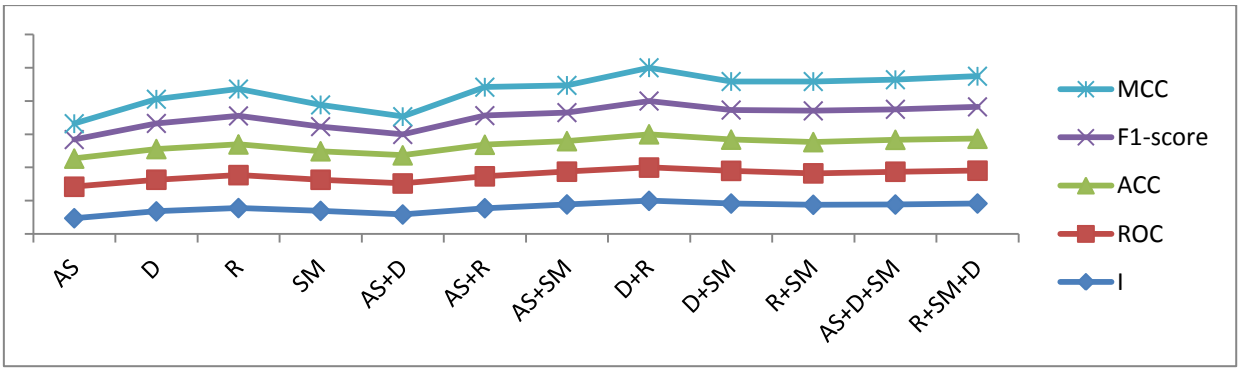

Figure 2. 80:20 split cross validation performance measures (MCC, F1-score, ACC, ROC and I) across 12 different pre-processing steps

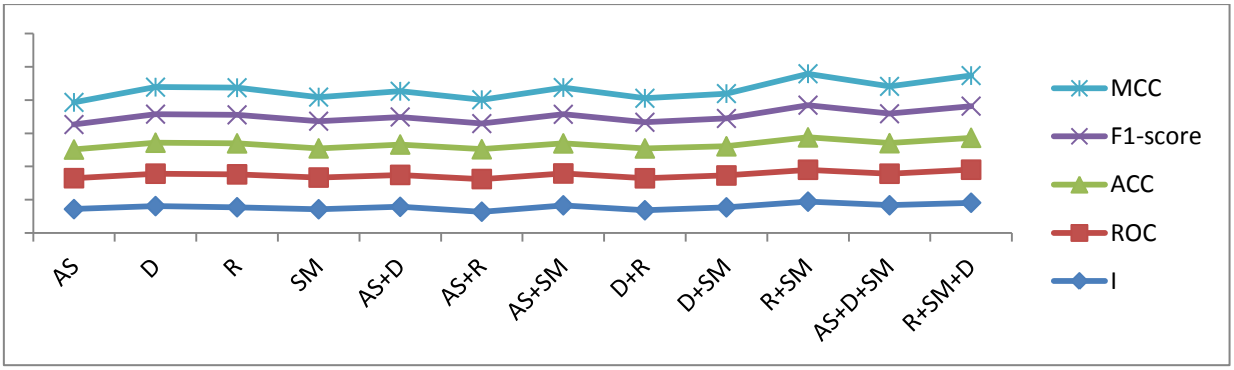

Figure 3. 70:30 split cross validation performance measures (MCC, F1-score, ACC, ROC and I) across 12 different pre-processing steps

Table 4: Performance measures (in %) of the MLP classifier on the 10 Fold Cross Validation PD dataset using different pre- processing technique

	K	I	ROC	TPR	TNR	PPV	NPV	FPR	FDR	FNR	ACC	F1	MCC	MK
AS	72.4	72.4	94.5	79.2	93.2	79.2	93.2	6.8	20.8	20.8	89.7	79.2	72.4	72.4
D	77.3	80.0	96.2	87.5	92.5	79.2	95.8	7.5	20.8	12.5	91.3	83.2	77.5	75.0
R	86.2	86.2	98.1	89.6	96.6	89.6	96.6	3.4	10.4	10.4	94.9	89.6	86.2	86.2
SM	83.8	84.5	97.0	92.7	91.8	88.1	95.1	8.2	11.9	7.3	92.2	90.4	83.9	83.2
AS+ D	78.8	77.2	95.5	81.3	95.9	86.7	94.0	4.1	13.3	18.8	92.3	83.9	78.9	80.7
AS+ R	77.0	76.1	92.3	81.8	94.3	84.9	93.0	5.7	15.1	18.2	90.8	83.3	77.0	77.9

AS+SM	83.0	83.9	95.6	92.7	91.2	87.3	95.0	8.8	12.7	7.3	91.8	89.9	83.1	82.3
D+R	**93.6**	**92.0**	**96.7**	92.7	99.3	98.1	97.2	0.7	1.9	7.3	97.4	**95.3**	**93.6**	95.3
D+SM	83.7	84.2	97.9	91.7	92.5	88.9	94.4	7.5	11.1	8.3	92.2	90.3	83.8	83.3
R+SM	88.6	88.6	96.9	93.6	95.0	93.6	95.0	5.0	6.4	6.4	94.4	93.6	88.6	88.6
AS+D+SM	80.2	80.4	96.9	88.5	91.8	87.6	92.5	8.2	12.4	11.5	90.5	88.1	80.2	80.1
R+SM+D	91.1	91.4	**98.8**	96.4	95.0	93.8	97.1	5.0	6.2	3.6	95.6	95.1	91.1	90.9

Table 5: Performance measures (in %) of the MLP classifier on the 80/20 split cross validation PD dataset using different pre- processing technique

	K	**I**	**ROC**	**TPR**	**TNR**	**PPV**	**NPV**	**FPR**	**FDR**	**FNR**	**ACC**	**F1**	**MCC**	**MK**
AS	47.8	47.8	94.6	57.1	90.6	57.1	90.6	9.4	42.9	42.9	84.6	57.1	47.8	47.8
D	72.3	68.3	95.1	71.4	96.9	83.3	93.9	3.1	16.7	28.6	92.3	76.9	72.6	77.3
R	80.5	78.2	99.0	81.8	96.4	90.0	93.1	3.6	10.0	18.2	92.3	85.7	80.6	83.1
SM	64.4	**69.8**	**93.5**	83.3	86.5	66.7	94.1	13.5	33.3	16.7	85.7	**74.1**	**65.1**	60.8
AS+D	53.0	58.9	93.3	71.4	87.5	55.6	93.3	12.5	44.4	28.6	84.6	62.5	53.7	48.9
AS+R	84.3	77.8	95.9	77.8	100.0	100.0	93.8	0.0	0.0	22.2	94.9	87.5	85.4	93.8
AS+SM	80.2	89.2	98.4	100.0	89.2	75.0	100.0	10.8	25.0	0.0	91.8	85.7	81.8	75.0
D+R	**100.0**	**100.0**	**100.0**	100.0	100.0	100.0	100.0	0.0	0.0	0.0	100.0	**100.0**	**100.0**	100.0
D+SM	84.7	91.9	98.2	100.0	91.9	80.0	100.0	8.1	20.0	0.0	93.9	88.9	85.7	800
R+SM	88.0	88.1	94.4	95.8	92.3	92.0	96.0	7.7	8.0	4.2	94.0	93.9	88.1	88.0
AS+D+SM	89.0	89.0	98.5	91.7	97.3	91.7	97.3	2.7	8.3	8.3	95.9	91.7	89.0	89.0
R+SM+D	**92.0**	**92.0**	**98.6**	95.8	96.2	95.8	96.2	3.8	4.2	4.2	96.0	**95.8**	92.0	92.0

Table 6: Performance measures (in %) of the MLP classifier on the 70/30 split cross validation PD dataset using different pre- processing technique

	K	**I**	**ROC**	**TPR**	**TNR**	**PPV**	**NPV**	**FPR**	**FDR**	**FNR**	**ACC**	**F1**	**MCC**	**MK**
AS	65.7	72.1	92.9	85.7	86.4	66.7	95.0	13.6	33.3	14.3	86.2	75.0	66.7	61.7
D	81.2	81.2	97.4	85.7	95.5	85.7	95.5	4.5	14.3	14.3	93.1	85.7	81.2	81.2
R	81.2	77.7	98.9	80.0	97.7	92.3	93.3	2.3	7.7	20.0	93.1	85.7	81.6	85.6
SM	72.4	**71.7**	95.3	80.0	91.7	83.3	89.8	8.3	16.7	20.0	87.7	81.6	**72.4**	73.1
AS+D	77.0	78.9	96.0	85.7	93.2	80.0	95.3	6.8	20.0	14.3	91.4	82.8	77.1	75.3

AS+R	70.5	64.3	98.1	66.7	97.7	90.9	89.4	2.3	9.1	33.3	89.7	76.9	71.9	80.3
AS+SM	79.7	83.5	95.7	96.0	87.5	80.0	97.7	12.5	20.0	4.0	90.4	87.3	80.5	77.7
D+R	71.8	68.7	96.1	73.3	95.3	84.6	91.1	4.7	15.4	26.7	89.7	78.6	72.1	75.7
D+SM	73.9	77.4	95.8	92.0	85.4	76.7	95.3	14.6	23.3	8.0	87.7	83.6	74.7	72.0
R+SM	**94.5**	**94.5**	**95.8**	96.8	97.7	96.8	97.7	2.3	3.2	3.2	97.3	**96.8**	**94.5**	94.5
AS+D+SM	82.1	83.7	94.8	92.0	91.7	85.2	95.7	8.3	14.8	8.0	91.8	88.5	82.2	80.8
R+SM+D	91.7	91.3	**99.2**	93.5	97.7	96.7	95.6	2.3	3.3	6.5	96.0	**95.1**	**91.7**	92.2

Using the confusion matrix obtained from each experiment various performance measures (see table 2) were computed. A total of 36 different experiments were performed that involved the application of 12 different pre-processing steps on the PD dataset, over three different cross-validation techniques (10 fold, 80:20 split, 70:30 split). The tables 4, 5, and 6 records the computed performance measures across 12 different pre-processing steps using different cross-validation approaches 10-fold, 80:20 split and 70:30 split respectively.

In the figures 1, 2 and 3 we only show the performance metrics namely MCC, F1-score, ACC, ROC and I. The MCC, F1-score and I are well-known for measuring the quality of the binary (two-class) classifiers. Here we determine a collection of the best pre-processing steps for the classification of the PD dataset using the MLP classifier based on the above mentioned five different performance measures.

From the 10-fold cross validation experiments we note that the combination of the pre-processing steps D and R *i.e.* $D + R$ resulted in the best values for the MCC (0.936), F1-score (0.953), ACC (0.974) and I (0.920) (see table 4). The ROC (0.967) however was 0.21 magnitude lesser than the pre-processing step $R + SM + D$ which recorded the highest *i.e.* ROC of 0.988 (refer to table 4). Across all the other four performance parameters both the pre-processing steps $D + R$ and $R + SM + D$ were comparable. Although the pre-processing step SMOTE *i.e.* SM alone recorded a high value for ROC (0.97) but its MCC (0.839) was significantly lower. Thus we conclude that, for 10-fold cross validation the best pre-processing step for the classification of the PD dataset using the ANN based MLP classifier is $D + R$. The second best is the pre-processing step $R + SM + D$.

For the 80:20 split the pre-processing step $D + R$ recorded 100% across all the performance parameters (See table 5). Similar to the observations in the 10-fold cross validation experiments we found that the pre-processing step $R + SM + D$ to be the second best with the ROC of 0.986, F1-score of 0.958, MCC and I of 0.92 for the 80:20 split. Though the pre-processing step SM recorded a high ROC (0.935), their MCC (0.651), I (0.698) and F1-score (0.741) were significantly lower (See table 5).

For the 70:30 split experiments we noticed that the pre-processing step $R + SM$ to be significantly better than the other pre-processing steps. For the $R + SM$ pre-processing step all the five performance parameters were ≥ 0.945 (See table 6). The pre-processing step $R + SM + D$ recorded the second best with the ROC of 0.992 and with the second best value for MCC (0.917) and F1-score (0.951) (See table 6). The SM pre-processing step alone recorded a significantly low value for MCC (0.724) and I (0.717). The pre-processing step AS consistently recorded $< 90\%$ accuracy across all the performance measures except for the ROC (0.929).

Based on the various observations in this study we can conclude that the best pre-processing steps for the PD dataset when classified using the ANN based MLP classifier are the combination of the individual pre-processing steps namely R, D, SM *i.e.* $D + R$, $R + SM$, and $R + SM + D$. This study highlights the fact that the combination of the pre-processing steps (R, D, SM) are much better than the individual parts.

Discretization is the most popular pre-processing step used for data exploration and data preparation in data mining. It is well-known that unless the continuous attributes in the dataset are discretized it is hard to solve the classification problem by any algorithms. However, across all the experiments in this study we found that the discretization method alone resulted in a high accuracy (0.91-0.93) and ROC (0.95-0.98) however the MCC and the F1-scores were not promising. Across all the experiments the range of values for the F1-score and the MCC were [0.70-0.85). Due to the imbalanced nature of the PD dataset the techniques such as Resample (R) and SMOTE (SM) are expected to improve the prediction accuracy of the minority class [12]. In our experiments we noted that the MCC (0.80-0.862) and F1-score (0.857-0.896) for R was significantly higher than the MCC and F1-score for D and SM. For D both the MCC and F1-score were within the range of 0.72 and 0.85 and for SM both the MCC and F1-score were within the range of 0.65 and 0.90. Therefore, it can be inferred that Resampling the PD dataset would be a better option compared to Discretization or SMOTE alone in order to improve the prediction accuracy of the MLP classifier.

5. RELATED WORK

Many studies have been performed on the classification of the PD dataset. All the studies were mainly focused on identifying a better classification algorithm for the PD dataset. Sriram *et. al.* in [7] have recommended the use of the SVM classifiers for the PD dataset. Shahbakhi in [8] used the Genetic Algorithm (GA) to extract the best features in the PD dataset. Using the SVM classifier Shahbakhi has reported an accuracy of 93.66% and 94.22% for an optimized 7 and 9 features in

the PD dataset. Gharehchopogh and Mohammadi in [4] have reported an accuracy of 93.22% for the PD dataset where 70% of the dataset was used for training the MLP classifier and the remaining 30% was used for testing. Khan in [2] performed feature selection on the PD dataset and has identified 10 relevant attributes in this dataset. Upon the application of the classifiers namely K-NN, Random Forest and Ada Boost, Khan has reported an accuracy of 90.26%, 87.17% and 88.72% respectively [2]. Khemphila and Boonjing in [6] have reported an accuracy of 91.45% and 80.769% on the training and validation PD datasets. On a similar dataset with the number of features reduced to 16 nos. they have observed an accuracy of 82.05% and 83.33% on the training and validation dataset respectively. Gil and Manuel in [9] have reported a high accuracy of the classification on the PD dataset using the MLP classifier. They have observed a sensitivity of 99.3% and NPV of 97.06% [9]. Das in [10] has reported an accuracy of 92.9% for classification in the PD dataset using the ANN classifier. Cagler *et. al.* in [11] have reported an accuracy of 93.55% for the classification of the PD dataset using the MLP classifier.

We have not come across any study similar to ours that provides a very comprehensive comparison of the classification accuracies in the PD dataset when different pre-processing steps are applied.

6. CONCLUSION AND FUTURE SCOPE

This study is intended to understand how the different types of pre-processing steps can affect the prediction accuracy of the classifier. In the process of classifying the PD dataset using the ANN based MLP classifier we observed a significantly high prediction accuracy when the dataset was pre-processed using both the Discretization and Resample technique, both in the case of 10-fold cross validation and 80:20 split. In the case of the 70:30 split we found that the combination of the pre-processing steps namely the Resampling and SMOTE on the PD dataset resulted towards the higher prediction accuracy using the MLP classifier. On 80:20 split of the pre-processed (Discretized and Resampled) PD dataset the ANN based MLP classifier achieved a 100% classification accuracy with F1-score and MCC being 100%.

Future work can be extended to understand if the pre-processing techniques Discretization, Resampling and SMOTE all combined, separately, or in any combination contributed towards higher prediction accuracy on the PD dataset across a variety of different supervised classifiers. Further down the line this study can be extended to different type of Medline datasets.

7. REFERENCES

[1] Ramani, G., and Sivagami, G. 2011. Parkinson Disease Classification using Data Mining Algorithms. International Journal of Computer Applications, 32(9).

[2] Khan, S. U. 2015. Classification of Parkinson's Disease Using Data Mining Techniques. J. Parkinsons Dis Alzheimer Dis, 2(1)

[3] Parkinson's Disease Foundation. Retrieved from http://www.pdf.org/about_pd, retrieved on August 15, 2016.

[4] Gharehchopogh, F. S., and Mohammadi, P. 2013. A Case Study of Parkinson's disease Diagnosis using Artificial Neural Networks. International Journal of Computer Applications. 73(19).

[5] Parkinsons Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Parkinsons, retrieved on August 15, 2016.

[6] Khemphila, A., and Boonjing, V. 2012. Parkinsons Disease Classification using Neural Network and Feature selection. International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering. 6(4).

[7] Sriram, T. V. S., Rao, M. V., Narayana, G. V. S., and Khaladhar, D. S. V. G. K. 2013. Intelligent Parkinson disease prediction using machine learning algorithms. International Journal of Engineering and Innovative Technology. 3(3). 212-215.

[8] Shahbakhi, M. 2014. Speech Analysis for Diagnosis of Parkinson's Disease using Genetic Algorithm and Support Vector Machine. Journal of Biomedical Science and Engineering.

[9] Gil, D., and Manuel, D. 2009. Diagnosing Parkinson by using Artificial Neural Networks and Support Vector Machines. Global Journal of Computer Science and Technology. 9(4). 63-71.

[10] Das, R. 2010. A comparison of multiple classification methods for diagnosis of Parkinson disease. Expert Systems with Applications. 37(2). 1568-1572.

[11] Cagler, M. F., Cetisli, B., and Toprak, I. B. 2010. Automatic Recognition of Parkinson's Disease from Sustained Phonation Tests Using ANN and Adaptive Neuro-Fuzzy Classifier. Journal of Engineering Science and Design. 1(2). 59-64.

[12] Burnaev, E., Erofeev, P., and Papanov, A. 2015. Influence of Resampling on Accuracy of Imbalanced Classification. Eight Interntaional Conference on Machine Vision (ICMV 2015). 9875.