Comparision of Dimension Reduction Methods to Predict Walking Speed from Electroencephalography Data

Allen Rahrooh
Department of Statistics and Data Science
University of Central Florida
Orlando, Florida
rahrooan@knights.ucf.edu

Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract. **(***Abstract***)**

Keywords—Electroencephalography, Classification

I. Introduction & Background

A. Electroencephalography

Humans process data every second using different body parts such as the brain and based on the type of the task other parts of the body can react such as increased heart rate or rapid eye movement. An example of a task would be listening to music where a person might dance if they like the music which is first processed by the ears then to the brain to tell the body to dance which triggers rapid eye movement and increased heart rate. These brain signals function by activating an electric potential called an electroencephalogram (EEG). The EEG signals originate from the brain and are made of cells called neurons. These neurons then release neurotransmitters throughout the body based on the type of signal received by the brain such as eye movement, heart rate, or pain in the body. The release of the neurotransmitter causes an electrical polarity change, which is how the EEG is measured in milli Volts (mV) either non-invasively or invasively depending on the task. This brings up an interesting detail about EEG, which contains a mixture of body signals such as eye movement (ECoG), muscle activity (EMG), brain activity (EEG), and heart activity (ECG) which make it difficult to collect and analyze. In attempt to minimize the amount of non EEG signals researchers [1] [2] have recorded non movement activities such as taking a test or looking at different images, which require the subject to stay still to avoid the inference of other signals. In the recent years researchers [3] [4] [5] have been interested to collect EEG during movement tasks such as walking and attempted to extract the pure EEG signal which has yet to be done. These data collections are a lot more complex because the EEG electrode can pick up various electrical body and artificial signals such as someone talking or a door closing. Past researchers [5] have attempted to study EEG during walking to see how the brain functions at different speeds such as 0.5 m/s and 1.25 m/s or different degrees of incline and decline ranging from -10 to 10 degrees. From these walking studies researchers [4] [7] believe that there is a correlation walking speeds and how the brain functions. With this correlation it can help medical professionals better diagnose people with diseases like Parkinson's where a person can lose motor control and can cause a loss of balance and freezing of gait [8]. Like when a person is running the EEG signal will be high because more parts of the body are moving as compared to walking. With these movement studies another issue arises being that EEG can have a very high temporal dimension upwards of hundreds of thousands data points and it can be challenging to separate out the EEG signal from the other signals such as movement artifact, ECoG, and EMG.

B. Dimension Reduction

With the high temporal dimension of EEG researchers have used temporal dimension reduction methods such as Independent Component Analysis (ICA) and Principal Component Analysis (PCA) [2] [5] [7] in attempt to extract the pure EEG signal. An ICA seeks to extract independent source components where as a PCA seeks to extract the uncorrelated sources [9] [10].

C. Independent Component Analysis

ICA is a mathematical method for dimension reduction and extracting the source signals. ICA follows assumptions that the variables are mixtures of latent variables which are variables that are not observed directly and are independent of each other. These latent variables are also known as the independent components [10] and using the ICA method can extract the independent components of interest. The ICA comes from (1)

$$(X_1, \cdots, X_m)^T = a_{ij}(S_1, \cdots, S_m)^T \quad (1)$$

where $(X_1, \cdots, X_m)^T$ is a $m \ x \ 1$ random vector of signals, a_{ij} is a unknown constant and square matrix of size $m \ x \ m$, and $(S_1, \cdots, S_m)^T$ is a $m \ x \ 1$ vector of m unknown source signals [10]. ICA is good for EEG because it can extract and separate the source signals like movement artifact, EMG, and EEG. The performance of the source extraction using ICA depends

©2019 IEEE

on the quality of the data and the statistical characteristics of the sources and the mixture of data. For ICA there are two methods being temporal and spatial ICA (tICA and sICA). tICA uses the assumption that data points have an independence in time and is based on the "cocktail" party analogy in attempt to extract sources from the mixture. Researchers have used tICA and were able to partially separate motion artifact from EEG signals [5]. With the sICA it assumes spatial independent components where the temporal dimension is reduced [3].

D. Machine Learning

In this paper we attempt to classify and predict walking speeds based on the EEG signals using machine learning. Machine learning is method of data analysis that can be used for making predictions and decision making. For machine learning the data is split into a training and test set the most common split is 80% for training and 20% for testing but these can vary depending on the data. For this paper I used various machine learning methods such as ensemble, conditional probability, regression, and hyperplane methods.

E. Ensemble Model

An ensemble model utilizes uses multiple learning algorithms to make a final decision rather than just one learning algorithm. The common ensemble models used are bagging, boosting, and random forest. Bagging is known as bootstrap aggregation. This is where the algorithm will take samples from the training dataset with replacement. Once the desired number of resamples are generated classification trees are used to calculate the classification rate these classification rates combined using an equally weighted average for a final majority vote. Boosting is similar to bagging but the difference is that boosting will take the misclassified data points and put a heavier weight on those to "boost" the classification rate. At the end a weighted average is taken with heavier weights given on the best performing resampled data points. The last ensemble method is random forest which only uses classification trees to make prediction. A decision tree will break down the data until a "stump" is reached where a final decision is made.

F. Logistic Regression Model

The logistic regression model uses (2)

$$Y = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n}} \quad (2)$$

Where $\beta_0, \beta_1, \cdots \beta_n$ are the coefficients of each independent variable $(x_1, x_2, \cdots x_n)$ used to estimate the correlation between the dependent categorical variable Y and the independent continuous variable X. The logistic regression can either be binomial (Yes or No) where the dependent variable has two categories or multinomial where the dependent variable has more than two categories (Adult, Teen, Child)

G. Conditional Probability Model

The conditional probability model that is commonly used is called Naïve Bayes. Naïve Bayes is based on Bayes' Rule (3)

$$P(\mathrm{A} \mid \mathrm{B}) = \frac{P(B \mid A) * P(A)}{P(B)} \quad (3)$$

Which assumes conditional probability when Y = j that there is a corresponding predictor vector x_0 [11]. From the conditional probability a Bayes' Decision Boundary can be generated as shown in Fig. 1 where the boundary separates the different classes.

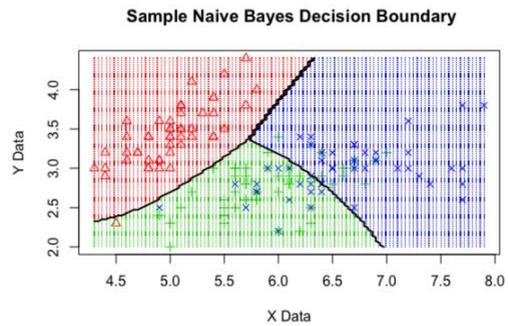

Fig. 1 Sample Bayes' Decision Boundary (Red, Green, and Blue Classes)

H. Hyperplane Model

The last machine learning model uses a hyperplane for decision making. This hyperplane is known as a Support Vector Machine as shown in Fig. 2. The hyperplane separates the training data set $\{x_1 \cdots x_n\}$ by assigning labels $\{y_1 \cdots y_n\}$ *where* $y_i \in \{-1,1\}$. The closest points that lie to the hyperplane are called the support vectors. For the hyperplane there is an inductive which builds a decision on the training data set and a transductive, which builds a decision on the training and test sets. To tune the hyperplane a function called a kernel is used. The most commonly used kernels for machine learning are the linear (4) and radial basis function (5) kernel.

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \quad (4)$$

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2\right) \quad (5)$$

The radial basis function kernel uses the parameters of γ and cost (c). The γ controls the decision boundary where high values can lead to a low margin but has a possibility of overfitting and a low γ can lead to a high margin with the possibility of misclassifying points. The c sets a penalty on the misclassified data points but can lead to smaller margins for high c values and larger margins for low c values.

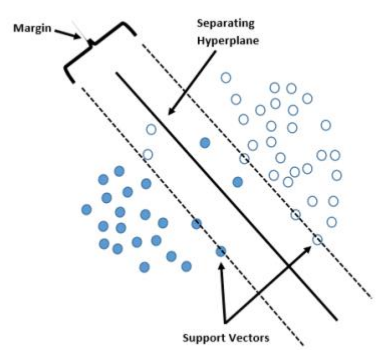

Fig. 2 Generated Data with Support Vector Machine Separation

II. METHODS

A. EEG Data Collection

We used a 128 EEG channel electrode system to collect data on young adults ages 18 – 35. They were instructed to walk on a treadmill at five different walking speeds being 0.5 m/s, 0.75 m/s, 1.0 m/s, 1.25 m/s, and a self-paced speed at level ground, incline and decline. The self-paced speed is based on the subjects location in space, which is measured by motion capture markers. If the subject moves forward then the treadmill will speed up and if the subject moves backwards the treadmill will slow down to keep the subject in the middle of the treadmill. Each trial lasted 5 minutes. The 128 EEG electrodes where placed on a cap based on cortical location. This data was collected for a study conducted by my colleague and later used for my thesis, which focused on the level ground data [20]. For this paper we will focus on comparing dimension reduction methods for one subject due to time constraints.

B. Pre-Processing Data

Once the raw EEG data was collected and exported we high pass filtered it at 1 Hertz (Hz) using MATLAB. The data was then imported into R as a .csv file. Each trial was 5 minutes (300 seconds) long with a sampling frequency of 512 samples per second which on average is 150,000 temporal data points. I decided to take the middle 50,000 subset because at the beginning the subject is speeding up and at the end the subject is slowing down so there would be a higher variance in the speed. I then attached speed labels to each data set being Speed 1 (0.5 m/s), Speed 2 (0.75 m/s), Speed 3 (1.0 m/s), Speed 4 (1.25 m/s), and Self-Paced. Each 50,000 subset for each speed was then combined row wise for a final dimension of 640 (128 per speed) x 50,001.

C. Dimension Reduction

Before the dimension reduction was performed I split the data using 80% for training and 20% for testing. The training set would consist of 515 EEG channels (103 per speed) and the testing set would consist of 125 EEG channels (25 per speed). The sICA was performed using the eegkit package [12] which reduced the data from 50,000 to 25 temporal points for a final training dimension of 515 x 25. From the sICA I extracted the W matrix, which maximizes the measure of normality, and calculated the matrix cross product with the testing data set to obtain a source matrix for the testing data which reduced the dimension from 125 x 50,000 to 125 x 25. The other dimension reduction method performed was randomly selected data points from the 50,000 temporal subset without running any ICA algorithm. So the training and testing sets ended up having the same dimensions as the ICA reduced data.

D. Classification Models

Once the ICA and no ICA dimension reduction methods were done I ran the used the training data set to train the model using the following classification models: Bagging, Boosting, Random Forest, Naïve Bayes, Multinomial Logistic Regression, and Support Vector Machines with a Linear and Radial Basis Function kernels. The following R packages were used for the classification models tree [13], randomForest [14], adabag [15], naivebayes [16], nnet [17], ISLR [18], and e1071 [19]. The support vector machines with a linear kernel had a cost set at 0.01 and the radial basis function kernel had a cost of 0.5 and γ of 0.1. These parameters can be tuned based on classification results.

E. Confusion Tables

Once the classification models are trained I took my testing data from the actual EEG and the predicted EEG and formed 5 x 5 confusion tables to analyze the performance of each classification model.

Confusion Table		
Predicted	Actual	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Fig. 3 General Confusion Table

Shown in Fig. 3 is a generalized confusion table where a True Positive (TP) represents a correctly classified positive observation. A False Positive (FP) represents a falsely classified positive observation. A False Negative (FN) represents a falsely classified negative observation. A True Negative (TN) represents a correctly classified negative observation. From the confusion tables the following performance metrics were calculated:

$$\text{Accuracy} = \frac{TP+FN}{TP+FP+FN+TN} \qquad \text{F1 Score} = 2 * \frac{\text{Precision x Sensitivity}}{\text{Precision} + \text{Sensitivity}} \qquad \text{Specificity} = \frac{TN}{TN+FP}$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Negative Predictive Value (NPV)} = \frac{TN}{TN+FN} \qquad \text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \qquad \text{False Negative Rate (FNR)} = \frac{FN}{FN+TP}$$

Fig. 4 Classification Performance Metrics

The accuracy measures how accurate the model is classifying the actual classes. The precision or positive predicted value (PPV) measures the positive classification performance. The sensitivity measures how well the positive class are correctly identified. The F1 score represents the weighted average of precision and sensitivity. The specificity measures how well the positive negative class are correctly identified. The NPV measures the true negative classification performance. The FNR measures the false negative classification performance. The FPR measures the false positive classification performance. For accuracy, precision, sensitivity, specificity, F1 score, and NPV the closer the value is to 1 the better the model is. For FNR and FPR the closer the value is to 0 the better the model is.

III. RESULTS

A. ICA Confusion Tables

Naïve Bayes					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	23	0	0	0	0
Speed 2	1	25	0	1	0
Speed 3	0	0	25	0	0
Speed 4	0	0	0	24	0
Speed 5	1	0	0	0	25

Multinomial Logistic Regression					
Predicited	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	17	3	0	0	0
Speed 2	0	23	0	0	0
Speed 3	0	0	25	0	0
Speed 4	2	9	0	9	0
Speed 5	1	3	0	0	25

Support Vector Machine Linear Kernel					
Predicited	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	24	1	4	3	0
Speed 2	0	24	1	4	0
Speed 3	0	0	20	0	0
Speed 4	1	0	0	18	0
Speed 5	0	0	0	0	25

Bagging					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	24	0	0	0	0
Speed 2	0	25	0	1	0
Speed 3	0	0	25	0	0
Speed 4	0	0	0	24	0
Speed 5	1	0	0	0	25

Boosting					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	22	0	0	0	0
Speed 2	0	25	0	1	0
Speed 3	3	0	24	0	0
Speed 4	0	0	0	23	0
Speed 5	0	0	1	0	24

Random Forest					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	24	0	0	0	0
Speed 2	0	24	0	1	0
Speed 3	0	0	25	0	0
Speed 4	0	0	0	24	0
Speed 5	1	1	0	0	25

Support Vector Machine Radial Basis Function Kernel					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	24	0	0	0	0
Speed 2	0	24	0	1	0
Speed 3	0	0	25	0	0
Speed 4	0	0	0	24	0
Speed 5	1	1	0	0	25

Fig. 5 ICA Classification Confusion Tables

In Fig. 1 are the classification confusion tables for the dimension reduced data using ICA. The darker the diagonal entries the better the performance is for each classification method. We see that overall the classification models predicted the walking speeds accurately.

B. ICA Classification Performance Metrics

ICA Performance Metrics of Classifiers				
Classifiers	Accuracy	Precision (PPV)	Sensitivity (Recall)	F1 Score
Naïve Bayes	0.98	0.98	0.98	0.98
Logistic Regression	0.85	0.83	0.89	0.83
SVM RBF	0.98	0.98	0.98	0.98
Boosting	0.96	0.96	0.96	0.96
Bagging	0.98	0.98	0.98	0.98
SVM Linear	0.89	0.90	0.89	0.89
Random Forest	0.98	0.98	0.98	0.98

ICA Performance Metrics of Classifiers				
Classifiers	Specificity	NPV	FNR	FPR
Naïve Bayes	0.99	0.99	0.02	0.01
Logistic Regression	0.96	0.96	0.11	0.04
SVM RBF	0.99	0.99	0.02	0.01
Boosting	0.99	0.99	0.04	0.01
Bagging	1.00	1.00	0.02	0.00
SVM Linear	0.97	0.97	0.11	0.03
Random Forest	0.99	0.99	0.02	0.01

Figure 6 ICA Classification Performance Metrics

In Fig. 2 are the calculated classification performance metrics for the data reduced with ICA, and the values are very good with the accuracy, precision, sensitivity, F1 Score, Specificity, and NPV being close to 1 and the FNR and FPR being close to 0.

C. No ICA Confusion Tables

Naïve Bayes					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	21	4	4	1	3
Speed 2	0	18	1	0	1
Speed 3	0	1	22	2	1
Speed 4	0	0	3	23	5
Speed 5	0	0	0	0	18

Multinomial Logistic Regression					
Predicited	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	20	2	2	0	1
Speed 2	0	20	1	1	1
Speed 3	0	1	26	1	0
Speed 4	1	0	0	24	1
Speed 5	0	0	1	0	25

Support Vector Machine Linear Kernel					
Predicited	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	20	0	0	0	0
Speed 2	0	23	1	0	0
Speed 3	0	0	27	0	1
Speed 4	1	0	1	26	0
Speed 5	0	0	1	0	27

Bagging					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	21	0	0	0	0
Speed 2	0	23	0	0	0
Speed 3	0	0	30	2	0
Speed 4	0	0	0	26	1
Speed 5	0	0	0	0	27

Boosting					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	21	0	0	0	0
Speed 2	0	23	0	0	0
Speed 3	0	0	30	0	0
Speed 4	0	0	0	26	1
Speed 5	0	0	0	0	27

Random Forest					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	21	0	0	0	0
Speed 2	0	23	0	0	0
Speed 3	0	0	30	0	0
Speed 4	0	0	0	26	1
Speed 5	0	0	0	0	27

Support Vector Machine Radial Basis Function Kernel					
Predicted	Actual				
	Speed 1	Speed 2	Speed 3	Speed 4	Speed 5
Speed 1	21	0	0	0	0
Speed 2	0	22	1	0	0
Speed 3	0	1	29	2	2
Speed 4	0	0	0	24	0
Speed 5	0	0	0	0	26

Figure 7 No ICA Classification Confusion Tables

In Fig. 3 are the classification confusion tables for dimension reduced data without ICA applied. As in Fig. 1 the darker the diagonal entries the better the performance is for each classification method. We see that overall the classification models predicted the walking speeds accurately.

D. *No ICA Classification Performance Metrics*

Performance Metrics of Classifiers				
Classifiers	Accuracy	Precision (PPV)	Sensitivity (Recall)	F1 Score
Naïve Bayes	0.80	0.82	0.81	0.80
Logistic Regression	0.90	0.90	0.90	0.90
SVM RBF	0.95	0.96	0.95	0.96
Boosting	0.99	0.99	0.99	0.99
Bagging	0.98	0.98	0.98	0.98
SVM Linear	0.96	0.96	0.96	0.96
Random Forest	0.99	0.99	0.99	0.99
Performance Metrics of Classifiers				
Classifiers	Specificity	NPV	FNR	FPR
Naïve Bayes	0.94	0.94	0.19	0.06
Logistic Regression	0.97	0.97	0.10	0.03
SVM RBF	0.99	0.99	0.05	0.01
Boosting	1.00	1.00	0.01	0.00
Bagging	0.99	0.99	0.02	0.01
SVM Linear	0.99	0.99	0.04	0.01
Random Forest	1.00	1.00	0.01	0.00

Figure 8 No ICA Classification Performance Metrics

In Fig. 4 are the calculated classification performance metrics for the data reduced without ICA and as in Fig. 2 the accuracy, precision, sensitivity, F1 score, specificity, and NPV are close to 1 and the FNR and FPR are close to 0.

E. *Visual Comparison of Classification Performance Metrics*

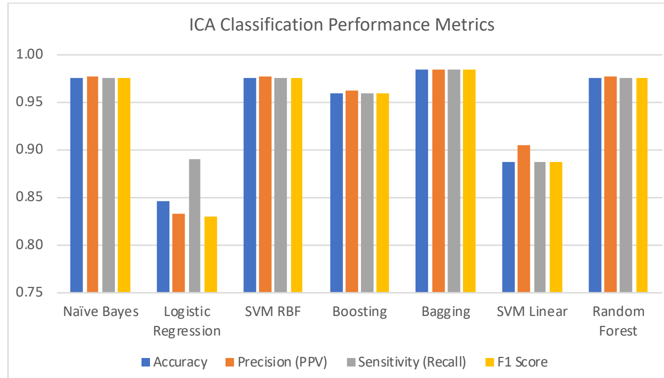

Figure 9 ICA Classification Performance Metrics for Accuracy, Precision, Sensitivity, and F1 Score

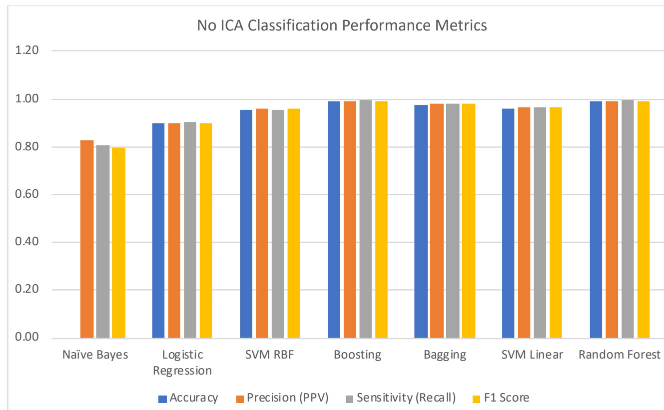

Figure 10 No ICA Classification Performance Metrics for Accuracy, Precision, Sensitivity, and F1 Score

We see that in Fig. 5 that the logistic regression and support vector machines with a linear kernel had the lowest performance metrics for the ICA reduced data and for the reduced data without ICA in Fig. 6 the Naïve Bayes had the lowest performance metrics.

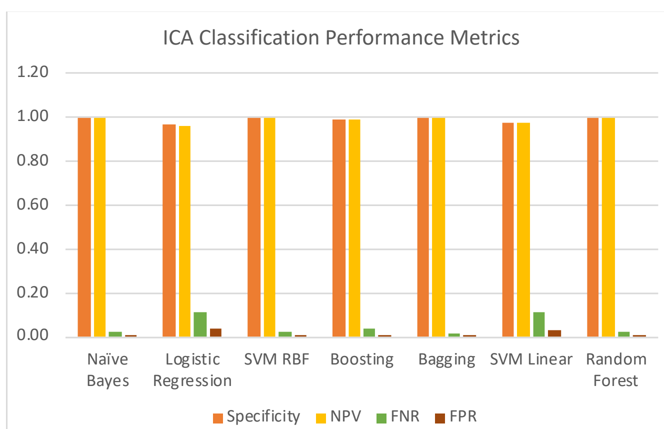

Figure 11 ICA Classification Performance Metrics for Specificity, NPV, FNR, and FPR

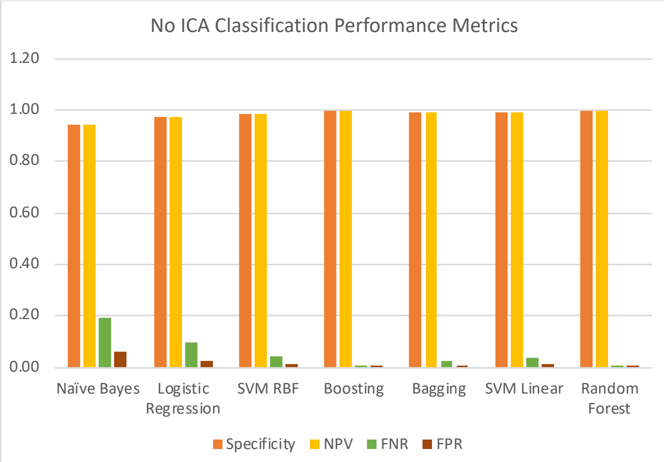

Figure 12 No ICA Classification Performance Metrics for Specificity, NPV, FNR, and FPR

We see that in Fig. 7 for the ICA reduced data that the logistic regression and support vector machines with a linear kernel performed the lowest with higher FNR and FPR values. In Fig. 8 we see that the Naïve Bayes performed the lowest with high values of FNR and FPR for the reduced data without ICA.

IV. Discussion

We see that for using ICA that the overall performance for the accuracy, precision, sensitivity, F1 score, specificity, and NPV ranged from 0.83 – 1.0 and for FNR and FPR the values ranged from 0 to 0.11. In comparison, without using ICA the overall performance for the accuracy, precision, sensitivity, F1 score, specificity, and NPV ranged from 0.80 – 1.0 and for FNR and FPR the values ranged from 0 – 0.19. This means that randomly selecting temporal features can have the same performance as doing an ICA for temporal dimension reduction. This means that using a less intensive dimension reduction method does not significantly impair the classification performance metrics. Also running ICA in RStudio takes hours to run on high dimensional data as compared to randomly selecting temporal features where it takes a few minutes. It also implies that there is a correlation between a humans brain activity and the gait cycle [4] [7] since we achieved high classification performance metrics.

V. Future work & Conclusions

For future analysis I can look at using other dimension reduction algorithms such as PCA, factor analysis, backward and forward feature selection, and clustering and then evaluating the same classification methods to see if any differences arise. Another factor to consider is the high pass filter applied in MATLAB we can use a more intensive filter to get a pure EEG signal that is free of other signals. Also, tuning the parameters of gamma and cost for the linear and radial basis function support vector machines to see if we can get a near perfect classification since they performed well with the ICA and without ICA. I can also apply the dimension reduction algorithm to analyze different mobile tasks such as collecting EEG while the subject is running or riding a bike to see how well the classification methods can perform on noisy data, since more brain activity is required to run or ride a bike then walking. I can also look into other classification methods such as neural networks, deep learning, ridge and elastic net regression, k nearest neighbors, and other kernels for the support vector machines. Overall the performance of the dimension reduction algorithm without ICA did not impair the classification performance metrics as compared to using a ICA.

Acknowledgment

I would like to thank the University of Central Florida BRaIN lab for providing the dataset and my past advisors for mentoring me on how to implement different classification and dimension reduction methods. The data collection was funded by Gerontology Research Award from the Learning Institute for Elders (LIFE) @ UCF & NIH R01AG064521.

References

[1] Y. P. Lin, C. H. Wang, T. L. Wu, S. K. Jeng, and J. H. Chen, “Support vector machine for EEG signal classification during listening to emotional music,” in Proceedings of the 2008 IEEE 10th Workshop on Multimedia Signal Processing, MMSP 2008, 2008.

[2] F. Artoni, A. Delorme, and S. Makeig, “Applying dimension reduction to EEG data by Prin- cipal Component Analysis reduces the quality of its subsequent Independent Component decomposition,” Neuroimage, 2018.

[3] Huang HH, Condor A, and Huang H. (2019) Classification of EEG Motion Artifact Signals Using Spatial ICA. Statistical modeling for biomedical research: contemporary topics in the field, Springer, 2019. Accepted.

[4] A. S. Oliveira, B. R. Schlink, W. D. Hairston, P. König, and D. P. Ferris, “Induction and separation of motion artifacts in EEG data using a mobile phantom head device,” J. Neural Eng., 2016.

[5] K.L. Snyder, J.E.Kline, H.J.Huang, and D.P.Ferris, “Independent Component Analysis of Gait-Related Movement Artifact Recorded using EEG Electrodes during Treadmill Walking,” Front. Hum. Neurosci., 2015.

[6] A. D. Nordin, W. D. Hairston, and D. P. Ferris, “Dual-electrode motion artifact cancellation for mobile electroencephalography,” J. Neural Eng., 2018.

[7] J. T. Gwin, K. Gramann, S. Makeig, and D. P. Ferris, “Removal of Movement Artifact From High-Density EEG Recorded During Walking and Running,” J. Neurophysiol., 2010.

[8] A.M.A.Handojosenoetal., “An EEG study of turning freeze in Parkinson’s disease patients: The alteration of brain dynamic on the motor and visual cortex,” in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2015.

[9] Holland, S. (2008). Principal Components Analysis. strata.uga.edu.

[10] Karhunen, J., Hyvarinen, A. and Oja, E. (2001). Independent Component Analysis. New York: John Wiley & Sons, Inc.

[11] J. Machado, A. Balbinot, and A. Schuck, “A study of the Naive Bayes classifier for analyzing imaginary movement EEG signals using the Periodogram as spectral estimator,” in ISSNIP Biosignals and Biorobotics Conference, BRC, 2013.

[12] N. E. Helwig and <helwig@umn Edu> Maintainer, “Package ‘eegkit’ Title Toolkit for Elec- troencephalography Data,” R Packag. version, 2015.

[13] Ripley, B. (2018) tree: Classification and Regression Trees. R package, 2018, version 1.0.39

[14] A. Liaw, M. Wiener, L. Breiman, and A. Cutler, “Package ‘ randomForest ’. Breiman and Cutler’s random forests for classification and regression,” Package “randomForest.” 2012.

[15] Cortes, Esteban Alfaro, Matias Gamez Martinez, and Noelia Garcia Rubio. adabag: Applies Adaboost.M1 and Bagging. R package, 2018, version 4.2.

[16] Majka, M. (2017) naivebayes: High Performance Implementation of the Naive Bayes Algo- rithm. R package, 2017, version 0.9.1.

[17] W. N. Venables and B. D. Ripley, "nnet: Feed-forward Neural Networks and Multinomial Log-Linear Models," R Packag., 2013.

[18] J. Gareth, W. Daniela, H. Trevor, and T. Rober, An Introduction to Statistical Learning with Applications in R. 2000.

[19] D. Meyer et al., "Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TUWien," The Comprehensive R Archive Network. 2017.

[20] Rahrooh, Allen, "Classifying and Predicting Walking Speed From Electroencephalography Data" (2019). Electronic Theses and Dissertations. 6281.