Predicting Alzheimer's Disease Progression: a Deep Learning approach

José Carlos André Nobre

Instituto Superior Técnico, Lisboa, Portugal

January 2021

Abstract

Alzheimer's Disease (AD) is a neurodegenerative condition that causes a deterioration in cognitive functions, affecting especially people of advanced age. As the disease is considered incurable, it is of the out most importance to follow the patients as earlier as possible. In particular as Mild Cognitive Impairment (MCI) is an early stage of Alzheimer's Disease, it is imperative to develop tools to allow predicting if and when a patient will progress from MCI to AD. Due to the recent rise of deep learning techniques and their great capabilities in terms of adaptability, this study will focus on the use of those machine learning methods to perform the prediction from MCI to AD. These will be allied with an approach using time windows, a method of dividing the data that besides giving the conversion of the patient can also predict when it will convert. A new methodology for Feature Selection (FS) based on Neural Networks was proposed as well as the use of a Missing Value Imputation (MVI) methodology based on autoencoders to create new data samples.

Keywords: Alzheimer's Disease, Mild Cognitive Impairment, Machine Learning, Deep Learning, Neural Networks, Prediction

1. Introduction

Neurodegenerative conditions affect mostly people of older age and lead to adverse effects on the patients as well as to their closest persons. The deterioration of the cognitive functions is a fact that no one can change, it affects a large part of the population at a certain point in their life, some of them can convert into Alzheimer if the deterioration is big and fast while others might only have a slight decline in cognitive functions. This creates a problem because if all people will suffer from neurological deterioration at some point in their life it is difficult to distinguish these signs from dementia and consequently Alzheimer's Disease (AD). A great loss of memory is usually one of the first indicators of AD but it might not be this straightforward all the time, that is why studies in this area of medicine have grown in the last decades. Better diagnosing techniques have appeared, such as Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), and also Neuropsychological Tests (NPT's). All these techniques have proven to help doctors in diagnosing this devastating disease.

This disease usually starts as a Mild Cognitive Impairment (MCI), which is a stage that lays between the cognitive decline of the usual aging process and a more serious decline of dementia. This impairment usually involves problems for the patient, like loss of memory, difficulties in language and thinking process that are more intense compared to the normal aging process of people. Consequently, if this impairment is not regarded carefully with the proper therapy, these patients can more easily progress into dementia, which in most cases turns to be AD. So, finding a way to predict if a patient will eventually suffer from Alzheimer will allow the medical staff to perform a better follow up of the patient. Also, it can give time and preparation to the patient's closest ones, which can give a more comfortable life to both.

Consequently, it is crucial to find methods to predict the progression of Alzheimer's Disease. Nowadays, new technologies revolve around Machine Learning. Due to the advances in this field and the amount of work dedicated to medical research with it ([20, 19, 24, 8, 6, 21, 17]). It certainly seems like the future of medical prediction is with machine learning techniques, these methods can help doctors narrowing the field of patients that can progress to AD and help to see if a patient will probably convert to AD in a given time. It is expected that these methods will evolve in the following years due to more computational power and the evolution of the machine learning, especially deep learning methods.

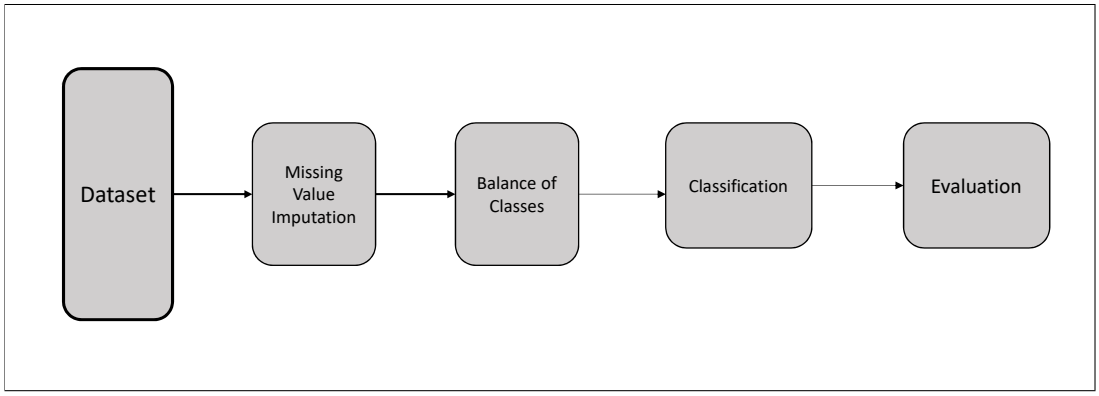

Figure 1: Typical classification workflow for medical data.

2. Database

The Database in which all the work was done is the Cognitive Complaints Cohort (CCC), created by a partnership of Santa Maria Hospital in Lisbon, the Laboratory of Language Studies, Memoclínica and the Neurology Department of Coimbra's University Hospital in order to investigate AD progression in patients with MCI, as stated in [10]. All the patients in this database are evaluated through Bateria de Lisboa para Avaliação das Demências (BLAD)[11], which is a neuropsychological battery validated for the Portuguese population. This database is composed of four different time windows (two, three, four and five years), these are databases on which the patients are grouped based on the information collected about the conversion or not to Alzheimer's Disease within a specific time window. The difference in this approach compared to the First Last Approach, which is the more usual database, is that the same patient can be classified with different labels in two different time windows. The patient can be Stable MCI (sMCI) in a smaller time window and Converter MCI (cMCI) in a larger window due to a later follow-up assessment where he can be diagnosed with AD. This allows us to know if and approximately when the patient will convert.

3. Background

In this section some of the methods used to predict the progression from MCI to AD will be explained, as well as the specific pipeline used to make this prediction. Starting with the machine learning pipeline to be used, in the case of the prediction of Alzheimer Disease's progression, because the dataset is not complete i.e., it has missing values, and has a large number of features, there is the need to consider a Feature Selection method and a missing value imputation method. Feature Selection is one of the major problems in this work because the features need to be well selected to have the better prediction possible. Since in some of the datasets the data is seriously unbalanced, there is also the need to balance the dataset. So, due to all of this, the proposed pipeline for the problem being dealt with is the one presented in Figure 1.

3.1. Missing Value Imputation

One of the problems that arise when working with large databases, especially in clinical databases, is the existence of missing values, due to patients missing appointments or not being able to take certain exams. This is a problem that can be dealt with in several different ways, and according to [5] the most common approaches to deal with missing data are:

- Ignore the features with missing values. If a feature has at least one missing value it is automatically ignored. This is not the best method for large datasets because the probability of having at least one missing value in each feature is very high, so we would be eliminating almost the entire dataset.
- Replace by the most common attribute value, in this, all the missing values are replaced by the most occurred value in the database. It is not the best in the case where we have categorical and numerical values in the same dataset.
- Replace by the most common attribute value in the class, which means that the most common value in a class will replace all the missing values in that feature. It is good for categorical data.
- The mean substitution. Here, the most mean value of the data in a feature is used to replace the missing values in that one. It is best suited for numerical data.
- Replace using regression or classification methods. In this approach, a classification or regression model is used to predict the values that will replace a missing attribute, it would

base the predictions on the remaining data in a class.

- Hot deck imputation. In this methodology, the missing values are replaced by similar cases in the database.

One of the new methodologies based on autoencoders to perform missing value imputation is the missing data importance-weighted autoencoder or MIWAE[16] which is based on the importance-weighted autoencoder(IWAE) [2]. The MIWAE goal is to fit a Deep Latent Variable Model (DLVM) into a dataset with missing data. DLVM's are latent variable models that use deep neural network architectures to ensure a higher flexibility on learning the underlying structure of data, such as clusters, patterns or statistical correlations. This type of models have problems when handling datasets with missing values. The usual methodologies when handling DLVM's such as variational autoencoders (VAE) or IWAE assume that the training data is fully observed, so MIWAE tries to overcome this limits. After training the DLVM using the MIWAE bound for missing data applications, this DLVM already knows the data distribution it can know fill the missing values with values that follow approximately the same distribution.

3.2. Data Balance Techniques

Most of the data in the real-world are imbalanced by nature. This situation occurs when the distribution of the target class (prediction) is not uniform among the different classes. This subject has revealed a lot of interest among the Machine Learning community because most of the Machine Learning Methods are created to work on a perfect dataset, this is a dataset where the classes are equally balanced. To overcome this class imbalance and improve the overall performance of the classifier there are 2 different types of techniques that can be used, undersampling and oversampling.

The undersampling method is a non-heuristic methodology, in which the database is reduced to obtain balanced classes, this means that we will remove instances of the database from the class with more instances to achieve the balance. There are 2 main methods of Undersampling [9], Random Undersampling(RUS) and Focused Undersampling(FUS), in the first method the instances from the majority class are randomly chosen to be removed to balance the classes, while in the second one, the instances of the majority class that are removed are the ones closest to the border between classes. Due to this reduction, the database will become smaller, which is not the best practice, because the training set will be less "rich" which can lead to a poorer classification. The upside of this method is that there is not the creation cases that are not real, which leads to a dataset with only real data.

The oversampling method is the opposite of the one stated before, here there is the creation of more cases to balance the classes, examples of techniques that use this approach are SMOTE[4], SMOTE-NC[4], Random Oversampling[14], Adaptive Synthetic Sampling (ADASYN)[12] and techniques based on auto-encoders [25, 23]. The upside is that there are more samples, which is really good for the training of our classifiers. On the other hand, there was the generation of samples that are not real, which is not the perfect scenario.

SMOTE (Synthetic Minority Oversampling)[4] works by creating synthetic examples instead of over-sampling with replacement, also, this operation is performed in the feature space rather than on the data space. Following what is said in [4], the minority class is over-sampled by taking each instance of that class and introduces those synthetic examples along the imaginary lines that connect the k nearest neighbors from the minority class. The steps to generate the synthetic samples are:

1. Take the difference between the feature vector, also called sample, and the nearest neighbors;

2. Multiply that difference with a number between 0 and 1 and add it to the feature vector.

These steps will cause the creation of a sample that is a random point along the line that connects two of the samples from the minority class. Ultimately, this approach will force the decision region of the minority class to become more general.

SMOTE-NC is a variant of SMOTE that works with datasets that have both numerical and categorical data, this way we can have better synthetic samples for our dataset.

3.3. Feature Selection

One of the crucial problems in machine learning tasks is to separate the relevant features from the not so relevant in a dataset, this is called Feature Selection (FS)[1, 22]. This separation of features is very important because it allows the reduction of noise in the data we are using as stated in [19]. Also, by reducing the subset of features that are used, we reduce the classification model complexity, which in turn helps to prevent the over-fitting of the model. There are three main types of Feature Selection methods: filter, wrapper and embedded. The first one evaluates feature worth based on the characteristics of the data and is independent of the machine learning algorithm, this is a filter method, so it does not need any classification algorithm associated with it in order to perform the selection.

Wrapper methods use the result obtained by a classification algorithm to see the importance of a subset of features. The last ones are a mix between feature selection and classification and the importance of feature is analyzed during the classification algorithm, one example is an L1 Regularization.

3.4. Classifiers

There are many classification methods that can be used for the prediction, such as Naïve Bayes, Support Vector Machines (SVM's) , Logistic Regression, K Nearest Neighbors (KNN), and Neural Networks. This work will focus on the latter.

Neural Networks (NN) [7] are based on the way that the human brain works, this gives them a great pattern recognition capability, which is why they are widely used on several problems such as image recognition tasks. A NN is composed by several perceptrons that are similar to human neurons, these are composed by summations and weight, the equation associated with these are shown in (1) and (2).

$$z_i = \sum_{j=1}^{k} w_{ij} x_{ij} + bias, \tag{1}$$

$$y_i = F(z_i). \tag{2}$$

In equation (1), w_{ij} is the weight from the connection between neuron i and j, and the bias is the bias from the neuron. The function denoted by $F()$ in (2), is called the activation function, this is a differentiable and function and it can be for example a sigmoid or a relu function. The goal of this function is to keep the output values of the network between certain values, in order for the network values not to raise indefinitely, this is why it can be also called squashing function.

The so-called Neural Network is nothing more than simply a set of those neurons organized in layers, as shown in Figure (2). There are three types of layers, the input, the hidden and the output layer. The first one is just the training instances from the dataset and the other two include neurons. In several layers case, the outputs from one hidden layer will be connected to the inputs from the next hidden layer, and the output layer will give the final result, the prediction. In binary classification cases we usually just have one perceptron in the output layer which will give either 0 or 1 for the predicted class, in the case where we have more classes, we will have one perceptron per class in the output layer.

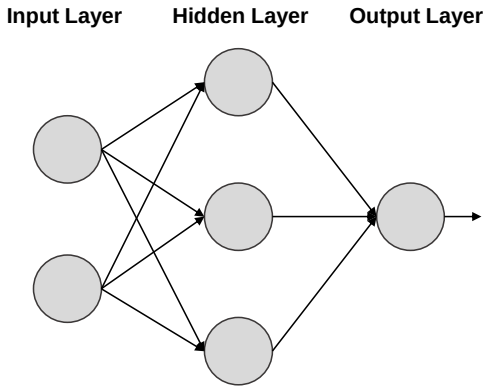

Figure 2: Multilayer Perceptron.

One important step from the neural networks which provides the learning capability is the Bakpropagation step, this step will compare the output value of the network with the real value provided by the user. It will make the comparison between values with the help from a loss function, for example the Mean Squared Error (MSE), after analyzing this error, the network will adjust the weights and bias of the perceptrons to reduce that error. These are made by differentiating the loss to each of the weights $\frac{\partial L}{\partial w_{ij}}$. After this propagation, when the first layer is reached, the network will update the weights to minimize the Loss.

One of the techniques that are being applied in Neural Networks to optimize and "simplify" these classifiers, reducing the processing power needed to train them and also make better predictions is the Pruning Technique [3, 13, 15, 18]. This type of technique is being widely used in heavy networks such as ResNet [18, 15], in order to reduce the training time and consequently provide researchers more time to tune and develop the network. In the works cited before, the results obtained with a large amount of pruning only slightly decreased the accuracy in some cases while in others the accuracy improved together with the training time. The most common approach of pruning [18] can be divided into two steps that happen on each epoch of the training of the network, in the first step after computing the gradient of the weights for the update, the importance of each of the neurons is analyzed using the average gradient, after this, the second step consists on removing the less important neurons on the network.

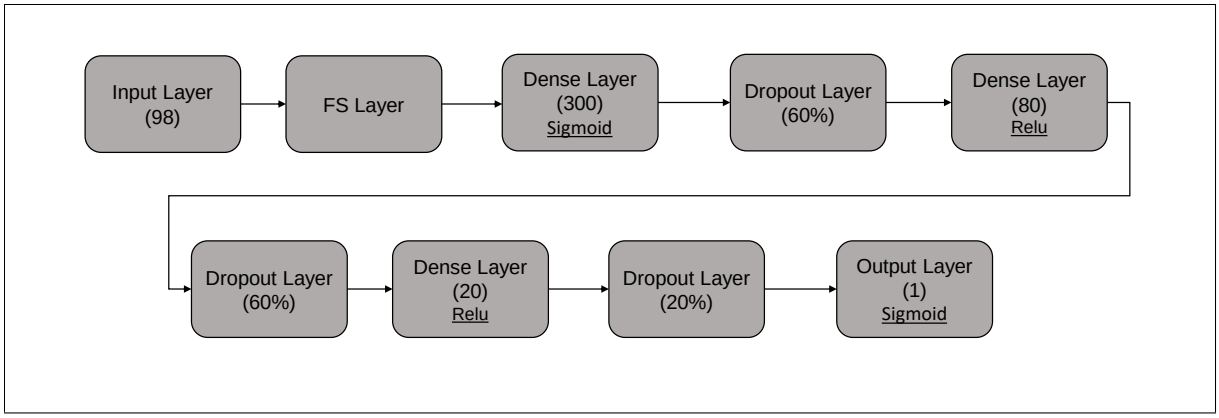

Figure 3: Neural Network Architecture.

3.5. Model Evaluation

When working on a classification problem, we need to find out which of the classifiers works best or if one is working at all on predicting the results, for this purpose there is the need to use model classification metrics. In this work, four of the most common metrics that are used in machine learning tasks and the medical environment were used: accuracy, Area Under the ROC Curve (AUC), sensitivity and specificity.

4. Implementation

For the purpose of this work a new methodology for Feature Selection (FS) based on Neural Networks was proposed as well as the use of a Missing Value Imputation (MVI) methodology based on autoencoders to create new data samples. This proposed methods will be explained in this section.

4.1. Layer Embedded Feature Selection

The Feature Selection methodology proposed for this work relies on Neural Networks to perform the selection of features. This method is based on the concept of pruning techniques and the goal is to eliminate the less useful features influence from the network to improve the overall accuracy of the results.

This methodology can be seen in Figure 4 where the first layer of the network is responsible to assign weights to the features. These weights are multiplied by the input features which are made to tend to zero by the loss function L.

$$L = MSE + \sum_{i=0}^{N} |w_i|, \tag{3}$$

So the loss function of the network will tend to zero, and consequently the weights will also tend to zero, this means that the features of the dataset f_i will be turned into f_i' in which

$$f_i' = w_i * f_i. \tag{4}$$

In order to make this method more adjustable a threshold t set by the user can be modified, and these weights w_i will be set to zero depending on this parameter t. This because neural networks do not naturally set the weights to zero, so this is done at the end of every batch on the training step, if a weight is below the threshold it will be set to zero, else it remains the same following the equation:

$$\begin{cases} 0, & \text{if } |w_i| < t \\ w_i, & \text{otherwise} \end{cases} \tag{5}$$

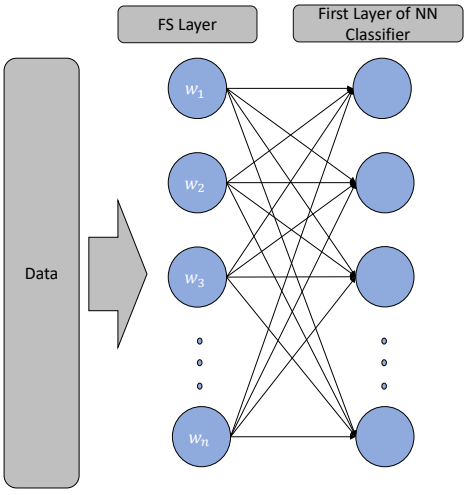

Figure 4: Layer Embedded Feature Selection.

This adjustment in the parameter t changes the amount of features chosen, if the parameter is increased the amount of features chosen will decrease, if t is decreased, the amount of features will increase. The benefits of using this method is that we do not need any independent feature selection algorithm as it is already embedded in the classifier, also as it uses neural networks, it has a high adapt-

ability to the datasets being used. The architecture of the Neural Network on which the FS method will be applied is the one presented on Figure 3. The activation functions used were the sigmoid for the first and last dense layers and relu for the two middle dense layers, the loss function used was the sum of the Mean Squared Error with the custom loss in equation 3.

4.2. MIWAE

The last methodology is using MIWAE to perform the oversampling. MIWAE was created only to perform the imputation of data, but one idea that arises in this thesis was that since the autoencoder already knows the data distribution from performing the missing value imputation, it could be used to generate new data to balance the classes.

As stated in the previous section, the model is built using a deep latent variable model, more specifically it is a DLVM with a Gaussian prior and a Student's t observation model.

$$p(x_i|z_i) = St(x_i|\mu_\theta(z_i), \Sigma_\theta(z_i), \nu_\theta(z_i)), \quad (6)$$

where μ_θ, Σ_θ, ν_θ are functions parametrised by the deep neural network, whose weights are stored in θ and x_i and z_i are respectively the data instances and the latent variables. After this, a decoder is built to support the three previous functions (μ_θ, Σ_θ, ν_θ), the encoder or inference network is then built using a Student's t approximation with an architecture that is similar to the decoder. Then the MIWAE bound is defined by the following equation:

$$\mathcal{L}_K(\theta, \gamma) = \sum_{i=1}^{n} E_{z_{i1},\ldots,z_{iK}\ q_\gamma(z|x_i^o)} \left[\frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x_i^o|z_{ik})p(z_{ik})}{q_\gamma(z_{ik}|x_i^o)} \right]. \quad (7)$$

Where p_θ is the posterior distribution, q_γ is the conditional distribution and p is the prior distribution.

The optimal imputation will be the conditional mean $E[x^m|x^o]$ that can be estimated with

$$E[x^m|x^o] \approx \sum_{l=1}^{L} w_l E[x^m|x^o, z_{(l)}] = \sum_{l=1}^{L} w_l \mu_\theta(z_l)^m \quad (8)$$

where

$$w_l = \frac{r_l}{r_1 + \ldots + r_L}, r_l = \frac{p_\theta(x_i^o|z_{(l)})p(z_{(l)})}{q_\gamma(z_{(l)}|x_i^o)} \quad (9)$$

By using the same procedure, we can use the same architecture to generate a whole row of data instead of only a few values for imputation,this is done by feeding the VAE with a empty row to evaluate instead of a row of data that only has some missing values, by estimating this one, a whole instance of a class can be produced.

With this methodology, the data generated follows a distribution that is close to the one from the original dataset, and with this more reliable results can be produced.

5. Results

Following the work previously done in [20, 19] using time windows, the first step was to create a simple pipeline shown in Figure 5, using the previously explained methods. To perform the validation of the algorithms four different datasets/time windows were used, from 2 to 5 years, and a 10-fold cross-validation methodology in each of the datasets to average the results.

To find the best combination of methodologies to predict the conversion from MCI to AD, an extensive number of tests was performed. This consisted of gradually replacing and testing each component (MVI, Data Balance, and FS) until obtaining the best combination.

By performing this methodology the best combination found was using MIWAE for Missing Value Imputation, SMOTE for data balance and to oversample the dataset and the proposed feature selection methodology allied with the corresponding Neural Network. The best results from this combination across the four time windows are shown in Table 1.

Table 1: Best results on the four time windows.

	ACC	AUC	SENS	SPEC
2 Year	0.780	0.822	0.677	0.811
3 Year	0.756	0.822	0.621	0.829
4 Year	0.766	0.855	0.695	0.832
5 Year	0.770	0.862	0.717	0.838

In Table 2 are the features chosen by our methodology which are common across all the time windows and across three time windows, these are the ones which are more important for the prediction. The features in bold are the ones which are also common with the work from Telma Pereira in [19].

When making a comparison between other classifiers such as Naïve Bayes, Support Vector Machines (SVM's) , Logistic Regression, K Nearest Neighbors (KNN) and FS methods (RFE, Sequential FS, Correlation), the overall best methodology is the Neural Network Architecture with the Feature Selection Layer proposed in this work. This has proved to be the best in all four time windows. While Naïve Bayes has proved to be one of the best in the work made by Telma Pereira in which this one is based [20, 19]. A comparison between the results obtained

Table 2: Most Common Feature selected by the proposed methodology.

Common Across All Datasets	Common Across 3 Datasets
PA_Dif_Total	DS_Forward
MVI_Free	LM_a_Total
MVI_Tot	***LM_a_Cued***
Orient_T	VisualM_B
Fluency_Sem	***Or_Total***
MPR_Total	Orient_P
a1_a5_Total	Proverb_Total
a_cr_int	MMS_Orientation_total
Depressao_GDS	MMS_OrientationTemporal_Total
MVI_Tot_Z	a_lg_int
Orient_T_Z	As_tot_Z
M_Initiative_Z	DS_back_Z
Proverb_Total_Z	TMT_B_temp_Z
LM_a_Total_Z	
LM_a_Interf_Z	

in this work and the ones obtained by Telma Pereira in [20, 19] can be seen in Table 3.

As seen in Table 3, the results obtained in this work are comparable with the ones obtained with the reference work that uses a complex FS ensemble to make the prediction. In terms of AUC on the 2 and 4 year windows, the results are very close to each other, with a noticeable difference in the 3 year window in which our methodology did not perform as well as the one in [19]. In matters of sensitivity, the results obtained here are not the best in comparison to the ones in the reference work, the same can not be said about the specificity values obtained, which were higher than the values obtained in [19] in every time window, this means that the capability of predicting the patients who will not convert to AD might be better.

6. Conclusions

The goal of this work was to predict the progression of Alzheimer's Disease, using machine learning methods. While in the search for a methodology that could bring us similar results to the ones obtained by [20], there was the idea to exploit the pruning techniques [3, 13, 15, 18] usually applied to reduce the complexity of deep learning models to feature selection. Naturally, Neural Networks have some learning capability to "reject" the less useful features by setting the associated weights close to zero. However, the idea is to improve that learning capability to reject features by eliminating them. Several classifiers (Naïve Bayes, Logistic Regression, Support Vector Machines, K Nearest Neighbors and Neural Networks) and different feature selection methods (Correlation, Recursive

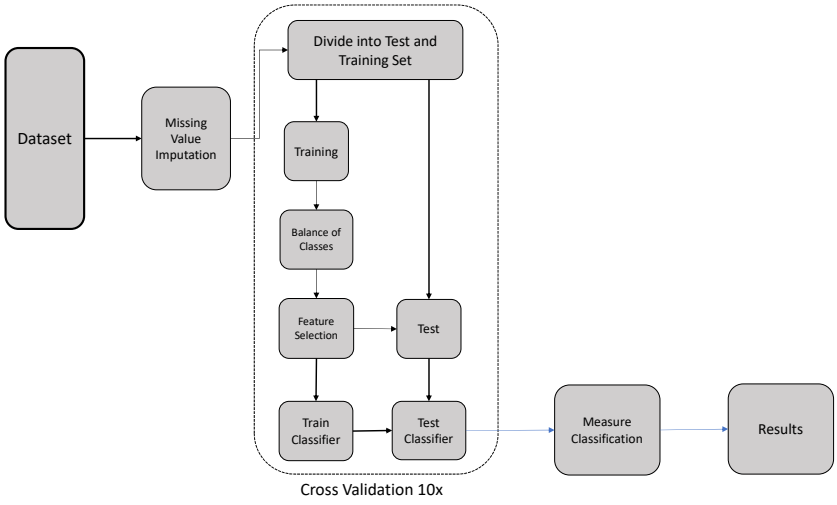

Figure 5: Pipeline Created for the prediction of AD Progression.

Table 3: Comparison between the results obtained in [19] with FS ensemble on the left and in this work on the right.

	AUC	SENS	SPEC
2 Year	0.821±0.00	0.738±0.02	0.765±0.01
3 Year	0.859±0.00	0.778±0.01	0.781±0.01
4 Year	0.868±0.00	0.793±0.01	0.788±0.00

	AUC	SENS	SPEC
2 Year	0.822±0.03	0.677±0.13	0.811±0.05
3 Year	0.822±0.03	0.621±0.11	0.829±0.08
4 Year	0.855±0.04	0.695±0.12	0.832±0.07

Feature Elimination and Sequential Feature Selection) were used to compare the proposed methodology. With these different methodologies, a plan was created to step by step combine them to find the overall best methodology for all the four time windows. First established a baseline methodology using MIWAE for missing value imputation, SMOTE as data balance and oversampling method, and the feature selection methodology allied with a deep neural network to perform the classification. In the first step, the missing value imputation method was changed making sure the other methods did not change, after finding out the best combination of MVI which was the MIWAE method the second step was to find the best data balance and oversampling method. After trying the four different methods the one which handled the best results was SMOTE, so this was the chosen method for balancing and sampling data. Finally, different Feature Selection methods were tested and the one who gave the best results was our methodology allied with the Neural Network. The overall results showed a better classification of our methodology in all but one time window, these results were also compared with the ones obtained in the work made by Telma Pereira et al. in [19]. In this comparison, our methodology gave similar results to the previously mentioned work in terms of AUC and higher specificity values, which means that the capability of predicting the patients who will not convert to AD might be better, but on the other hand, sensitivity results were lower than expected. These results have shown comparable capability of prediction to other state of the art works and capable of making predictions as early as 5 years before the conversion with accuracy values of 77%, sensitivity of 72%, specificity of 84% and ROC Area of 0.86.

References

[1] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[2] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[3] T. I. Burgess, K. Howard, E. Steel, and E. L. Barbour. To prune or not to prune; pruning induced decay in tropical sandalwood. *Forest ecology and management*, 430:204–218, 2018.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[5] G. Chhabra, V. Vashisht, and J. Ranjan. A classifier ensemble machine learning approach to improve efficiency for missing value imputation. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 23–27. IEEE, 2018.

[6] J. A. Cruz and D. S. Wishart. Applications of machine learning in cancer prediction and prognosis. *Cancer informatics*, 2:117693510600200030, 2006.

[7] G. Daniel. *Principles of artificial neural networks*, volume 7. World Scientific, 2013.

[8] O. M. Doyle, E. Westman, A. F. Marquand, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, S. C. Williams, et al. Predicting progression of alzheimer's disease using ordinal regression. *PloS one*, 9(8):e105542, 2014.

[9] T. Elhassan and M. Aljurf. Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. 2016.

[10] M. R. J. F. D. P. T. P. Florentino Fdez-Riverola, Mohd Saberi Mohamad. *11th International Conference on Practical Applications of Computational Biology Bioinformatics - 2017*. Springer, 2017.

[11] M. M. G. Guerreiro. *Contributo da neuropsicologia para o estudo das demências*. 1998.

[12] H. He, Y. Bai, E. A. Garcia, and S. Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE*

International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pages 1322–1328. IEEE, 2008.

[13] Q. Huang, K. Zhou, S. You, and U. Neumann. Learning to prune filters in convolutional neural networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 709–718. IEEE, 2018.

[14] A. Liu, J. Ghosh, and C. E. Martin. Generative oversampling for mining imbalanced datasets. In *DMIN*, pages 66–72, 2007.

[15] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

[16] P.-A. Mattei and J. Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, pages 4413–4423, 2019.

[17] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem. Classification using deep learning neural networks for brain tumors. *Future Computing and Informatics Journal*, 3(1):68–71, 2018.

[18] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019.

[19] T. Pereira, F. Ferreira, S. Cardoso, D. Silva, A. Mendonça, M. Guerreiro, and S. C . Madeira. Neuropsychological predictors of conversion from mild cognitive impairment to alzheimer's disease: A feature selection ensemble combining stability and predictability. *BMC Medical Informatics and Decision Making*, 18, 12 2018.

[20] T. Pereira, L. Lemos, S. Cardoso, D. Silva, A. Rodrigues, I. Santana, A. de Mendonça, M. Guerreiro, and S. C. Madeira. Predicting progression of mild cognitive impairment to dementia using neuropsychological data: a supervised learning approach using time windows. *BMC medical informatics and decision making*, 17(1):110, 2017.

[21] S. Sarraf and G. Tofighi. Classification of alzheimer's disease using fmri data and deep learning convolutional neural networks. *arXiv preprint arXiv:1603.08631*, 2016.

[22] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

[23] Z. Wan, Y. Zhang, and H. He. Variational autoencoder based synthetic data generation for imbalanced learning. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[24] J. Ye, M. Farnum, E. Yang, R. Verbeeck, V. Lobanov, N. Raghavan, G. Novak, A. DiBernardo, and V. A. Narayan. Sparse learning and stability selection for predicting MCI to ADs conversion using baseline ADNI data. *BMC neurology*, 12(1):46, 2012.

[25] Y. Zhang. Deep generative model for multi-class imbalanced learning. 2018.