http://dx.doi.org/10.35596/1729-7648-2023-21-6-106-112

Original paper

UDC 004.78; 615.47

USING MACHINE LEARNING FOR RECOGNITION OF ALZHEIMER'S DISEASE BASED ON TRANSCRIPTION INFORMATION

ULADZIMIR A. VISHNIAKOU, YU CHU YUE

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Submitted 12.07.2023

Abstract. The purpose of this article is to perform analytical and prognostic studies on the recognition of Alzheimer's disease based on decoded text speech data using machine learning algorithms. The data used in this article is taken from the ADReSS 2020 Challenge program, which contains speech data from patients with Alzheimer's disease and healthy people. The problem under study is a binary classification problem. First, the full texts of the interviewees were extracted from the transcribed texts of the speech data. This was followed by training the model based on vectorized text features using a random forest classifier, in which the authors used the GridSearchCV method to optimize hyperparameters. The classification accuracy of the model reached 85.2 %.

Keywords: machine learning, random forest method, binary classification, optimization parameters.

Conflict of interests. The authors declare no conflict of interests.

For citation. Vishniakou U. A., Yu Chu Yue (2023) Using Machine Learning for Recognition of Alzheimer's Disease Based on Transcription Information. *Doklady BGUIR*. 21 (6), 106–112. http://dx.doi.org/10.35596/1729-7648-2023-21-6-106-112.

ИСПОЛЬЗОВАНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ БОЛЕЗНИ АЛЬЦГЕЙМЕРА НА ОСНОВЕ ТРАНСКРИПЦИОННОЙ ИНФОРМАЦИИ

В. А. ВИШНЯКОВ, ЮЙ ЧУ ЮЭ

Белорусский государственный университет информатики и радиоэлектроники (г. Минск, Республика Беларусь)

Поступила в редакцию 12.07.2023

Аннотация. Выполнены аналитические и прогностические исследования по распознаванию болезни Альцгеймера на основе расшифрованных текстовых речевых данных с использованием алгоритмов машинного обучения. Данные были взяты из программы ADReSS 2020 Challenge, которая содержит речевые данные пациентов с болезнью Альцгеймера и здоровых людей. Распознавание болезни Альцгеймера представляет собой проблему бинарной классификации. Сначала из расшифрованных текстов речевых данных извлекались полные тексты интервьюируемых пациентов. Затем следовало обучение модели нейронной сети на основе векторизованных текстовых признаков с использованием классификатора случайного леса, в котором авторы применяли метод GridSearchCV для оптимизации гиперпараметров. Точность классификации модели составила 85,2 %.

Ключевые слова: машинное обучение, метод случайного леса, бинарная классификация, параметры оптимизации.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Вишняков, Ю. А. Использование машинного обучения для распознавания болезни Альцгеймера на основе транскрипционной информации / Ю. А. Вишняков, Юй Чу Юэ // Доклады БГУИР. 2023. Т. 21, № 6. С. 106–112. http://dx.doi.org/10.35596/1729-7648-2023-21-6-106-112.

Introduction

Alzheimer's disease (AD) is an insidious and progressive neurodegenerative disease, clinically defined as the impairment of certain cognitive and functional abilities [1]. Internationally, no medical treatment has been developed to cure AD, which causes progressive and irreversible damage to the patient, in all cases leading to neurological death [2]. Memory loss and language impairment are among the earliest symptoms of AD and have become the direction in which scholars are working to help Alzheimer's patients today with the development of artificial intelligence.

The development of the Internet of Things (IoT) has made it possible to ensure patients' safety by means of emergency buttons, global positioning systems, intelligent smoke detectors, etc. [3]. Advances in machine learning has made it possible for technology to analyze subtle differences in speech expressions and word frequency variations of patients. The pronunciation in AD is characterized by variations in different temporal and acoustic phonological parameters, with clinical symptoms of dysphasia, i. e., naming impairment, impaired auditory and written comprehension, fluent but hollow speech, and speaking with semantic errors [4]. Therefore, during the conversation they gave less specific information relative to healthy controls and the syntax of the sentences used was simpler, these characteristics were the basis for conducting the presented research.

Review

In recent years, many scholars have attempted to conduct computational analysis of verbal and linguistic disorders in AD. Fraser et al. [5] extracted variables from narrative speech of Alzheimer's patients in an exploratory analysis, selected over 370 available features to construct a model and achieved 81.92 % classification accuracy in distinguishing patients from healthy controls. Gábor Gosztolya [6] created a set of acoustic features based on eight acoustic markers: intelligibility, voice rhythm, speech length, duration of voiceless and filled pauses (hesitation), number of voiceless and filled pauses as well as hesitation rate, trying to combine them with linguistic features, including morphological features, speech-based spontaneous features and semantic features to distinguish healthy controls from those with different stages of dementia. In their paper, the accuracy of the phonological features were compared separately from the semantic features as well. In the 2-class machine learning task, an accuracy value of 86 % was obtained by combining the "extended" set of acoustic features with all linguistic features in the task of differentiating controls from patients with mild AD.

In this article, the authors used machine learning to identify AD by vectorizing the transcribed text of speech data to achieve a distinction between Alzheimer's patients and healthy control participants using TfidfVectorizer [7], a random forest classifier, and GridSearchCV.

ADReSS Challenge dataset

The authors' data came from the ADReSS 2020 Challenge dataset [8], whose principal target is to address the lack of standardization that is currently affecting the field by introducing a dataset in which different methodologies can be systematically compared. The dataset has two parts, a training set and a test set, containing a total of 1955 speech segments from 78 non-patient participants and 2.122 speech segments from 78 AD participants, for each speech data the recordings was acoustically enhanced using fixed noise. Only the training set from the competition dataset was used in this experiment, containing 54 AD patients and 54 control participants, totaling 108 participants.

It elicited narrative speech using the "Cookie theft" picture description task from the Boston Diagnostic Aphasia Examination [9]. The procedure instructs the interviewer to show participants the picture and encourage them to describe it, available data contains complete enhanced audio and normalized

audio blocks. Only the transcribed text corresponding to the full augmented audio was used in this experiment. The transcribed text was provided by the dataset, they were annotated using the CHAT encoding system [10].

TfidfVectorizer feature extractor

TfidfVectorizer is an open source method for natural language text processing, whose central idea is the TF-IDF (Term frequency – inverse document frequency) text feature representation method [11], a common weighting technique for information retrieval and data mining. The main idea of TF-IDF is that if a word or phrase appears in an article with high frequency and rarely appears in other articles, it is considered to be highly relevant to the document, which means that the word or phrase has good category differentiation ability that is suitable for classification.

The TfidfVectorizer uses the inverse domain frequency (IDF) and term frequency (TF) of the words to compute their corresponding term frequency inverse domain frequency (TF-IDF) values. The TF of word t in a given text corpus is calculated as the formula below:

$$tf(t,d) = count(t,d),$$

where t is a term; d is the given text corpus.

The result of $count(t, d)$ is the number of occurrences of term t in document d. Whereas the IDF of word t in d is calculated as formula below:

$$idf(t) = \log\left(\frac{N}{df(t)}\right),$$

where N refers to the total number of texts in document; $df(t)$ is the number of texts containing the term t.

The Tf-idf of the word t in the given document d is computed using the equations given below:

$$Tf_{idf(t,d)} = tf(t,d) \cdot idf(t),$$

where $tf(t, d)$ is the term frequency of word t in document d, which denotes the number of times t appeared in document d divided by total number of words in the document.

Random forest classifier

Random forest is an algorithm that integrates multiple trees through the idea of ensemble learning, its basic unit is the decision tree. Specifically, each decision tree is a classifier, for an input sample, N trees will have N classification results, random forest designates the category with the highest number votes as the final output, one of the simplest bagging ideas. It can reduce the variance in the decision tree, has good robustness against noise and outliers [12], having the characteristics of being able to handle high-dimensional data with high accuracy also.

The training process of the random forest classifier is as following:

1) samples from the training set are randomly sampled to form a new training set (bagging method, Bootstraping);

2) randomly select some features from the training set to form a new feature set;

3) a decision tree is trained based on the new training set and new feature set. The decision tree is trained by continuously dividing the dataset into smaller subsets until the number of subsets is so small that it is somehow predefined or it cannot be divided any further;

4) repeat steps 1 to 3 to train multiple decision trees for forming a forest.

Fig. 1 shows the relationship between decision trees and random forest.

Since the parameters of decision trees in random forests need to be selected manually, which can have a significant impact on the performance of the model if they are not selected properly for the hyperparameters, GridSearchCV was used in this article to do this task of searching. GridSearchCV is a variant of GridSearch, which is an algorithm evaluation method for model selection and parameter tuning using cross-validation to select the best hyperparameters. Specifically, GridSearchCV divides hyperparameter space into subspaces and performs cross-validation on each subspace to select the best hyperparameter combination.

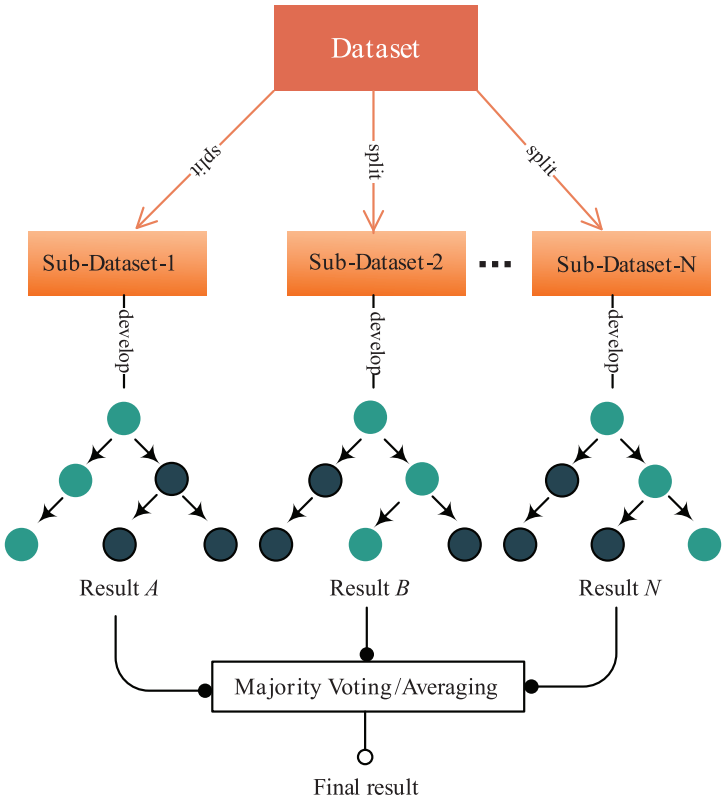

Fig. 1. The flow of the random forest classifier

It can improve stability and generalization ability of the model, as well as reduce the risk of overfitting. In this experiment, the number of folds for cross-validation of GridSearchCV method was set to 10, the number of parallel running jobs was set to 6 to accelerate the grid search process.

Experiments and methodology

The dataset language for this experiment is English, the programming language is python. Python is one of the most popular programming languages in the field of machine learning, which not only has rich machine learning libraries such as scikit-learn, TensorFlow, PyTorch, Keras, etc. to help practitioners quickly build machine learning models, but also has powerful visualization capabilities to visualize and display data.

After the transcribed texts corresponding to the complete enhanced audio were obtained, the authors first extracted all the speech text passages related to the participants in the control and AD patient groups by file separately after cleaning, integrated the speech text passages in each file into a string, which were then organized in the form of rows and columns using the DataFrame class in the Pandas library to form a table for subsequent referencing and analysis. In order to distinguish the files of control and patient groups, the authors added a column “ad” marker to identify groups to which the files belong. Finally, the control and patient files were combined and disrupted to return a complete processed dataset named “train_df” in code.

After the preparation of dataset, the authors adopt the *K*-fold cross-validation procedure technique to divide the dataset into 10 subsets, which were divided into training dataset and test dataset in the ratio of 8:2. After the segmentation, four variables were obtained, named as “train_features, test_features, train_labels and test_labels”.

The authors combined TfidfVectorizer and random forest classifier into a whole workflow using Pipeline class within scikit-learn library, trained with train_features and train_labels of training set as input, the best parameters of TfidfVectorizer and random forest classifier obtained after GridSearchCV method search were shown in Tab. 1.

Table 1. Parameters setting for TfidfVectorizer and random forest classifier

TfidfVectorizer parameters	Setting	Random forest classifier parameters	Setting
vec__max_features	2000	clf__n_estimators	10
vec__stop_words	'english'	clf__max_depth	10
vec__analyzer	'word'	clf__min_samples_split	5
vec__max_df	0.5	clf__min_samples_leaf	2
vec__sublinear_tf	True	clf__bootstrap	True

The meanings of the parameters of the text feature extractor in Tab. 1 are as follows: *Vec_max_features* specifies the maximum number of different words or characters allowed as features; *vec_stop_words* is used to determine whether to remove stop words, which are frequently occurring but generally insignificant words in text analysis; *vec_analyzer* is used to determine whether the features are based on words or characters; *vec_max_df* limits the maximum document frequency allowed for a word, words with frequencies exceeding this threshold will be removed; *vec_ sublinear_tf* is used to specify whether or not to use sublinear TF scaling, which means that the word frequencies of the text data should be compressed or scaled.

The meanings of each parameter in the random forest classifier in Tab. 1 are as follows: *clf_n_estimators* controls the number of decision trees in the random forest, as the number of trees increases, the model's complexity also rises; *clf_max_depth* sets the maximum depth of each decision tree, a larger depth may lead to overfitting, while a smaller depth may result in underfitting; *clf_min_samples_split* specifies the minimum number of samples required to split a node in a decision tree; *clf_min_samples_leaf* defines the minimum number of samples required to be at a leaf node, using a smaller value will lead to fewer samples at the leaf nodes; *clf_bootstrap* is a binary parameter that controls whether bootstrap sampling is used during training of each decision tree, when it is set to True, the training data for each tree is randomly sampled with replacement.

The parameters of the text feature extractor in Tab. 1 were: maximum number of features retained by the feature extractor was 2000, deactivated words were selected as English deactivated words, analysis was performed according to words, words with a frequency threshold of more than 0.5 in the feature extractor would be ignored, and sublinear scaling was selected for the scaling of word frequencies. The parameters of the random forest classifier in Tab. 1 were: the number of decision trees contained in the constructed random forest was 10, maximum depth of each decision tree was 10 layers, minimum number of samples required for decision tree splitting was 5, minimum number of samples required on leaf nodes was 2, and bagging method with put-back was used.

The performance of the model was evaluated using the test set after the model was trained. Since the data in the dataset were labeled data, the problem studied by the authors was the classification problem of supervised learning, Fig. 2 shows the specific experimental procedure conducted in this article, where the input Train_df was the name of the processed dataset.

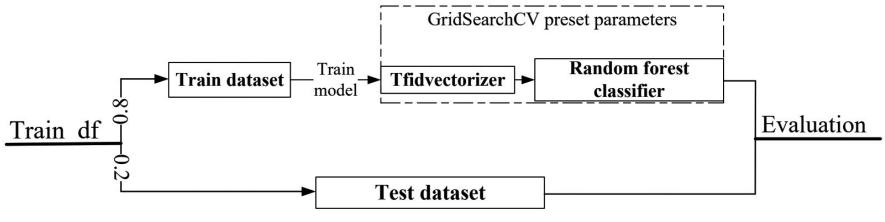

Fig. 2. Experimental flowchart

Results discussion

To effectively and comprehensively evaluate the proposed model, the authors adopted three resampling methods, respectively namely: *K*-fold cross validation, Leave one subject out (LOSO) cross validation and bootstrap sampling method. Since for small sample datasets, more information can be obtained through repeated sampling.

1. *K*-fold cross validation

Cross validation is a procedure for validating a model's performance without replacement. *K*-fold cross validation means that the original dataset is divided into pre-specified number *K* mutually exclusive subsets (usually evenly divided) called "folds", these subsets are produced in a systematic way and then the model is trained and tested in K iterations. Each time the model uses *K*-1 of the folds as training set, the remaining one would be used as validation set, each of fold would been left out exactly once. Since in this experiment 10 *n_folds (n_splits)* were requested, the authors received 10 iterations and it gave 10 different accuracies, the authors took the average of them as performance estimation. *K*-fold cross validation can utilise dataset more efficiently as all the data is used, also the estimation error is reduced to some extent as multiple training and testing are used, but in some cases cross validation can suffer from bias or variance.

2. Leave one subject out (LOSO) cross validation

LOSO is a specific cross-validation method tailored to the subject-based nature of the dataset in this experiment. It is a variation of *K*-fold cross-validation, where *K* equals the number of subjects in the dataset. During each iteration, LOSO utilizes one subject as the test set while training the model on data from all other subjects. This approach effectively assesses the model's generalization to new subjects, but it comes with heightened computational costs, particularly when dealing with a large number of subjects in the dataset.

3. Bootstrap sampling method

The bootstrap method is a resampling technique applied in machine learning to estimate the skill of machine learning models when making predictions on data not included in the training data. It is a statistical method and allows one to calculate confidence intervals for the results of cross-validation to obtain a range of confidence in the model's performance. Due to the drawing with replacement, a new bootstrapped sample data set may contain multiple instances of the same original cases or completely omit other original cases, although 2 sample sets are unlikely to be 100 % same, but it may still introduce sampling bias, since it can not fully simulate the true distribution of the dataset. Advantages of this method is that it is easy to implement and can be effective in providing a plausible range of model accuracies.

Here the authors set method as new sample sets with the same number of cases as the original data set, because the sample size of the original dataset is small, if the number of samples in each bootstrap sampling is less than the number of samples in the original dataset, it affects the model's generalisation ability and stability. The authors set iteration number as 5, in each bootstrap iteration, sample would create one model, which is tested against the Out of Bag (test data) of that sample, thus obtained accuracies for 5 samples, the final accuracy output is the average of them.

Tab. 2 presents a comparison of the accuracy achieved by various methods for evaluating the models proposed in this article, alongside the baseline result from reference [8]. The baseline result was obtained using the LDA classifier with linguistic features on LOSO cross-validation for the AD classification task.

Table 2. Compare with ADReSS challenge baseline result

Datasets	Researchers	Research methods	Evaluation method	Accuracy, %
Ad_speech	Luz S. [8]	LDA classifier	LOSO CV	77.0
Ad_speech	Authors	Random forest classifier	LOSO CV	85.2
Ad_speech	Authors	Random forest classifier	*K*-fold CV	87.6
Ad_speech	Authors	Random forest classifier	Bootstrap Sampling	87.3
Ad_speech	Yuan J. [13]	ERNIE model	LOO (leave-one-out)	89.6

IIn addition, based on bootstrap sample calculations, the authors are sure that the true accuracy of the model my be more 87.3 % on others data set. Tab. 2 shows that the results of this experiment surpassed the LDA classifier, slightly inferior to the ERNIE model and allowed to obtain a good classification for the Republic of Belarus.

Conclusion

This article presents a solution for the analysis and prediction of Alzheimer's disease using publicly available datasets from which the authors extracted complete transcripts of participants' speech.

The data set was divided and introduced into a machine learning model in which a random forest classifier was used to implement the task of recognizing Alzheimer's disease with optimized hyperparameters and using the GridSearchCV method. The results showed that the classification experiments showed an accuracy of 85.2 % (the code is stored in https://github.com/HkThinker/Using-Machine-Learning-for-Recognition-of-Alzheimer-s-Disease-Based-on-Transcription-Information/tree/main) exceeded the linguistic baseline provided for the ADReSS 2020 challenge, losing by 4 % to the ERNIE model, which means that the classification experiments achieved promising results for Belarus. By converting the transcription text into TF-IDF feature vectors using TfidfVectorizer in combination with a random forest classifier, these classification results were achieved for the task of classifying Alzheimer's disease.

References

1. Martínez-Sánchez F., Meilán J. J. G., Carro J., Ivanova O. (2018) A Prototype for the Voice Analysis Diagnosis of Alzheimer's Disease. *Journal of Alzheimer's Disease*. 64 (2), 473–481.
2. Pulido M. L. B., Hernández J. B. A., Ballester M. Á. F., González C. M. T., Mekyska J., Smékal Z. (2020) Alzheimer's Disease and Automatic Speech Analysis: A Review. *Expert Systems with Applications*. 150, 113213.
3. Lorenz K., Freddolino P. P., Comas-Herrera A., Knapp M., Damant J. (2019) Technology-Based Tools and Services for People with Dementia and Carers: Mapping Technology Onto the Dementia Care Pathway. *Dementia*. 18 (2), 725–741.
4. Szatloczki G., Hoffmann I., Vincze V., Kalman J., Pakaski M. (2015) Speaking in Alzheimer's Disease, is That an Early Sign? Importance of Changes in Language Abilities in Alzheimer's Disease. *Frontiers in Aging Neuroscience*. (7), 195.
5. Fraser K. C., Meltzer J. A., Rudzicz F. (2016) Linguistic Features Identify Alzheimer's Disease in Narrative Speech. *Journal of Alzheimer's Disease*. 49 (2), 407–422.
6. Gosztolya G., Vincze V., Tóth L. Pákáski M., Kálmán János, Hoffmann Ildikó (2019) Identifying Mild Cognitive Impairment and Mild Alzheimer's Disease Based on Spontaneous Speech Using ASR and Linguistic Features. *Computer Speech & Language*. 53, 181–197.
7. Garreta R., Moncecchi G. (2013) *Learning Scikit-Learn: Machine Learning in Python*. Great Britain, Packt Publ.
8. Luz S., Haider F., de la Fuente S., Fromm D., MacWhinney B. (2004) Alzheimer's Dementia Recognition Through Spontaneous Speech: The ADReSS Challenge. *arXiv Preprint arXiv*. 06833, 2172–2176.
9. Pompili A., Abad A., de Matos D. M. (2020) Pragmatic Aspects of Discourse Production for the Automatic Identification of Alzheimer's Disease. *IEEE Journal of Selected Topics in Signal Processing*. 14 (2), 261–271.
10. Brian MacWhinney (2014) *The CHILDES Project: Tools for Analyzing Talk. Vol. II: The Database*. Moscow, Psychology Publ. 432.
11. Manning C. D. (2009) *An Introduction to Information Retrieval*. Cambridge, Cambridge University Publ. 581.
12. Breiman L. (2001) Random Forests. *Machine Learning*. 45, 5–32.
13. Yuan J., Bian J., Cai X., Huang J, Ye Z., Church K. (2020) Disfluencies and Fine-Tuning Pre-Trained Language Models for Detection of Alzheimer's Disease. *Interspeech*. 2020, 2162–2166.

Authors' contribution

The authors contributed equally to the writing of the article.

Information about the authors

Vishniakou U. A., Dr. of Sci. (Tech.), Professor at the Department of Infocommunication Technologies of the Belarusian State University of Informatics and Radioelectroctronics

Yu Chu Yue, Postgraduate at the Department of Infocommunication Technologies of the Belarusian State University of Informatics and Radioelectroctronics

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovki St., 6
Belarusian State University
of Informatics and Radioelectronics
Tel.: +375 44 486-71-82
E-mail: vish@bsuir.by