An early aortic dissection screening model and applied research based on ensemble learning

Lijue Liu[1,2], Shiyang Tan[1], Yi Li[1,2], Jingmin Luo[3], Wei Zhang[3], Shihao Li[1]

[1]School Of Information Science And Engineering, Central South University, Changsha, China; [2]Hunan ZIXING Artificial Intelligence Research Institute, Changsha, China; [3]Xiangya Hospital, Central South University, Changsha, China

Contributions: (I) Conception and design: L Liu, S Tan, Y Li; (II) Administrative support: Y Li, J Luo, W Zhang; (III) Provision of study materials or patients: J Luo, W Zhang; (IV) Collection and assembly of data: S Tan, S Li; (V) Data analysis and interpretation: L Liu, S Tan; (VI) Manuscript writing: All authors; (VII) Final approval of manuscript: All authors.

Correspondence to: Yi Li. Central South University, Changsha 410008, China. Email: 1871347480@qq.com.

Background: As a particularly dangerous and rare cardiovascular disease, aortic dissection (AD) is characterized by complex and diverse symptoms and signs. In the early stage, the rate of misdiagnosis and missed diagnosis is relatively high. This study aimed to use machine learning technology to establish a fast and accurate screening model that requires only patients' routine examination data as input to obtain predictive results.

Methods: A retrospective analysis of the examination data and diagnosis results of 53,213 patients with cardiovascular disease was conducted. Among these samples, 802 samples had AD. Forty-two features were extracted from the patients' routine examination data to establish a prediction model. There were five ensemble learning models applied to explore the possibility of using machine learning methods to build screening models for AD, including AdaBoost, XGBoost, SmoteBagging, EasyEnsemble and XGBF. Among these, XGBF is an ensemble learning model that we propose to deal with the imbalance of the positive and negative samples. The seven-fold cross validation method was used to analyze and verify the performance of each model. Due to the imbalance of the samples, the evaluation indicators were sensitivity and specificity.

Results: Comparative experiments showed that the sensitivity of XGBF was 80.5%, which was better than the 16.1% of AdaBoost, 15.7% of XGBoost, 78.0% of SmoteBagging and 77.8% of EasyEnsemble. Additionally, XGBF had relatively high specificity, and the training time consumption was short. Based on these three indicators, XGBF performed best, and met the application requirements, which means through careful design, we can use machine learning technology to achieve early AD screening.

Conclusions: Through reasonable design, the ensemble learning method can be used to build an effective screening model. The XGBF has high practical application value for screening for AD.

Keywords: Aortic dissection (AD); early screening; machine learning; ensemble learning

Submitted Feb 13, 2020. Accepted for publication Oct 10, 2020.
doi: 10.21037/atm-20-1475
View this article at: http://dx.doi.org/10.21037/atm-20-1475

Introduction

Aortic dissection (AD) is a very dangerous cardiovascular disease. The main causes of AD are hypertension, Marfan syndrome and aortic atherosclerosis (1-3). The blood in the aortic cavity enters the arterial wall through the cracked intima that causes intimal separation from the medial membrane and hematoma. The hematoma mass is driven by high blood pressure and spreads along the long axis of the artery (4). AD usually results in high morbidity and mortality (5,6). Once AD is onset, it will quickly lead to death. The mortality rate within 24 hours after the onset is about 1% to 2%, 50% within 48 hours, and 60% to

70% within one week. Its five-year natural survival rate is only 10% to 15% (7-9). Due to the different locations in which tears can occur and the extent of the tear, patients' symptoms and signs are complex and diverse, making the rate of misdiagnosis and missed diagnosis reach 30% to 40% (10,11). Many patients miss the best treatment period for these reasons. It has been reported that 10.6% of patients with AD are misdiagnosed as having acute coronary syndrome (ACS) on first diagnosis (12). About 1% to 2% of patients with AD may develop acute myocardial infarction (AMI); however, AD and AMI are completely different in terms of treatment (13). Once AD patients get the wrong treatment, such as antithrombotic, thrombolytic, or emergency CAG/PCI, which are methods to treat AMI, all are associated with poor prognosis and increased risk of death due to AD. Therefore, if a patient is misdiagnosed or a diagnosis is missed, the patient is likely to be unable to obtain a further accurate diagnosis, or timely and accurate treatment. In China, the overall treatment level of AD is still low; the rate of missed diagnosis, hospitalization and preoperative mortality is comparatively high, and the age of patients is getting younger (14). Moreover, the development of medical resources and technology in China is imbalanced. In primary and underdeveloped hospitals, the lack of medical facilities and experienced doctors means higher rates of misdiagnosis and missed diagnosis, as well as higher mortality. Thus, a simple and effective early screening method is very necessary.

With the development of information technology and the popularity of electronic medical records, machine learning is constantly being applied to medical diagnosis to improve diagnostic accuracy, provide early prediction, reduce doctor pressure, and inspection costs. For example, Dwivedi (15) used six machine learning algorithms to assist in the diagnosis of ischemic heart disease. The best performance of his study was logistic regression with accuracy, sensitivity and specificity of 85%, 89% and 81%, respectively. Gatuha and Jiang (16) used machine learning algorithms to diagnosis breast cancer, and their best result had an accuracy of 97%. Liu *et al.* (17) established machine learning models to predict embryonic development. All of these methods have achieved better results than traditional methods. Machine learning diagnostic methods for AD have also attracted people's attention. Huo *et al.* used the Bayesian network, Naive Bayes, decision tree J48 and SVM algorithms to classify AD emergency patients (18). Their study contained 492 samples, including 330 patients with AD and 162 patients misdiagnosed as AD, which means the misdiagnosis rate reached 33%, and the sample ratio between AD patients and non-patients was close to 2:1. However, the purpose of their research was to find the misdiagnosed patients from the patients diagnosed with AD, rather than to screen out high-risk groups. The sample size was small. Liu *et al.* (19) analyzed the performance of several machine learning models in AD screening, among which the SmoteBagging was the best, and the sensitivity reached 78.1%. Wu *et al.* (20) used the Random Forest model to investigate the risk of in-hospital rupture in type A AD.

The purpose of our study is to explore if the patients' routine examination data can be used to establish a rapid early screening model to advise doctors or patients on whether further examination is required.

We present the following article in accordance with the TRIPOD reporting checklist (available at http://dx.doi.org/10.21037/atm-20-1475).

Methods

Clinical information

The study was conducted in accordance with the Declaration of Helsinki (as revised in 2013). The study was approved by the Ethics Board of Xiangya Hospital, Central South University (201502042). This study is a retrospective study, all data were desensitized data from hospital's electronic medical records, which did not contain patient identification information, and the consent was waived.

The registry data of 53,213 inpatients in the Cardiovascular Department of Xiangya Hospital from January 2008 to December 2016 were analyzed. There were 802 AD patients in this database, and the rest were hospitalized patients with other cardiovascular diseases including viral myocarditis, myocardial infarction and coronary heart disease. The diagnosis of AD was mainly based on medical imaging methods: (I) The Computed Tomography (CT) image showed one or more torn aortic intima and both true and false cavities could be found on the aortic; a series of complications may have been seen based on AD leakage or rupture, such as pericardial, mediastinal and pleural effusions, blood accumulation or aortic valve regurgitation. (II) Magnetic Resonance Imaging (MRI) showed differentiable high-signal true and false cavity images; in the Field Echo (FE) sequence scan image, both the true cavity and the false cavity may have been shown as high signals, and the low signal inner diaphragm could be seen between them; (III) Contrast agent overflow or ejection from aortic incision could be seen in the Computed Tomography Angiography (CTA); the shunt signals of the contrast agent dividing into two cavities with the blood flow could be observed; low-density linear endometrial flap could be seen between the true cavity and the false cavity. (IV) Incision and differentiable cavities could be directly observed in the intima or meniscus of the aortic during aortic surgery and postmortem of patients.

Model introduction

AD is a relatively rare cardiovascular disease, so the proportion of AD patients to non-AD patients is low, and the distribution of the two types of samples is extremely imbalanced. The purpose of the study is to screen for patients with AD from a large number of possible patients. Therefore, a major challenge of machine learning research for this work is how to deal with the problem caused by imbalance. Traditional prediction methods focus on global accuracy, and the formula is as follows:

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

Where *TP* is the number of true positives; *TN* is the number of true negatives; *P* is the number of positive samples; *N* is the number of negative samples, and *FP* is the number of false positives.

According to the formula, when *P*>>*N* or *N*>>*P*, even if none of the samples' predictions is correct, the accuracy can still be high. The imbalance causes the predictor to ignore the minority. This means that the predictor always prefers to discriminate some ambiguous samples into the majority class.

In view of the shortcomings of traditional algorithms, our study proposes an oversampling ensemble algorithm named Extreme Gradient Boosting Forest (XGBF). XGBF is an ensemble learning model which is composed of several XGBoost classifiers (21). The specific structure of the XGBF model is shown in *Figure 1*.

Since XGBoost has a fast convergence rate, the training speed of XGBF can be guaranteed. The training data entered into each XGBoost classifier were composed with some undersampled majority data and oversampled minority data. The oversampling operations included duplicating and Smote (22), so the learning model could get more information from the minority samples. The undersampling operation used non-replacement sampling to draw a certain amount of samples from the majority class set each time to make the distribution of positive and negative samples be similar. Finally, each weak classifier was enhanced by ensemble methods to achieve better predictions.

The XGBF algorithm combines the advantages of the oversampling, undersampling and ensemble methods so that the predictor can fully learn the characteristics of the samples and produce better results.

Cross-validation of predictors

We used seven-fold cross-validation to verify the stability of the classifier. Before the training and verification (cross-validation), we randomly divided the AD patient set and non-AD patient set into seven disjointed subsets of the same size. Then one AD subset and one non-AD subset were merged together to get seven new subsets. The training and testing procedures were repeated seven times. Each time, one of the seven subsets was picked as a test set, and the others were merged together as a training set. A total of 687 AD patients and 44,924 non-AD patients were included in six training sets, and the test data set included 115 AD patients and 7,487 non-AD patients. In order to avoid the accident of the experiment, this study evaluated the average of ten experiments. In different experiments, the training set and test set were randomly split, so they were different in each experiment.

Comparison methods and evaluation parameters

The experimental study applied four ensemble learning algorithms including AdaBoost, XGBoost, SmoteBagging and EasyEnsemble to compare with XGBF. The introduction of these algorithms is introduced in the Supplementary file 1. The computer configuration of these experiment was: 64-bit Windows10 OS, Python3.6, 16G RAM, and CPUi5-6500.

A confusion matrix, as shown in *Table 1*, was used to show some basic evaluation indicators. A sample can be divided into true positive cases (TP), false positive cases (FP), true negative cases (TN) and false negative cases (FN) according to the combination of its real category and prediction category. In our case, the positive sample refers to the minority class of patients with AD, and the negative sample refers to the majority class of patients with non-AD.

As we mentioned, the proportion of the positive and negative samples was extremely imbalanced. The traditional evaluation indicators, such as accuracy and

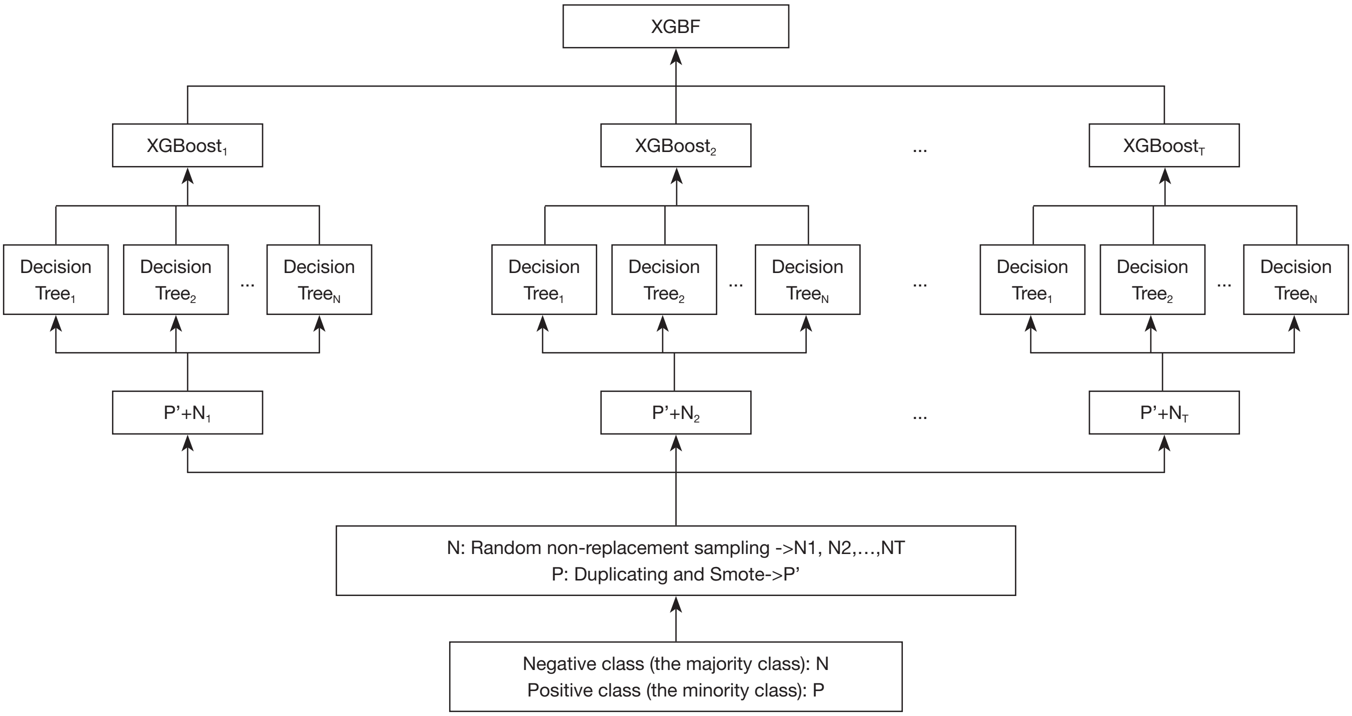

Figure 1 The specific structure of the XGBF model. This is the structure of XGBF model. In the algorithm, P represents positive (the minority class) dataset, which is the patient set; N represents negative (the majority class) dataset, which is the non-patient set; and T represents the number of XGBoost classifiers. The training data entered into each XGBoost classifier were composed with some undersampled majority data and oversampled minority data. In the XGBF algorithm, the oversampling operations were performed on the minority class, which included duplicating and smote. The minority samples were strengthened by these two methods, so the learning model could get more information from the minority samples. The undersampling operation was performed on the majority class so that the distribution of each kind of sample was balanced. Finally, each weak subclassifier was enhanced by ensemble methods to achieve better classification results.

Table 1 Confusion matrix

	Predicted positive class	Predicted negative class
Actual positive class	TP (True positive)	FN (False negative)
Actual negative class	FP (False positive)	TN (True negative)

positive predictive value (PPV), were no longer suitable. For example, there was a predictor that predicted all samples as majority class, while the accuracy was still high, this kind of predictor is meaningless in this study. The situation of PPV is similar. Even if we have a predictor that can accurately predict all minority classes and has a low false positive rate, it is still possible to get a lower PPV, because there are far more majority than minority classes, resulting in the number of FPs being greater than the number of TPs. Therefore, the evaluation indicators used in this study were sensitivity and specificity. After all, the purpose of screening is to find as many of the high-risk groups as possible. Compared with false positives, the risk of false negatives is higher.

Statistics

Python3.6 software was used for statistical analysis of the data in this study. Measurement data are expressed as mean ± standard deviation. Count data are a ratio or percentage. The differences in count data between the two groups were compared by the chi-squared test. The differences in measurement data were compared using the two-independent-sample *t* test. $P<0.05$ was considered statistically significant. The diagnostic performance of the classifiers was described using sensitivity and specificity.

Results

Study cohort

Our AD dataset was obtained from Xiangya Hospital of Central South University. All sample data were extracted from electronic medical records (EMR), including patients' information documents, hospitalization records and laboratory medical records. The information contained in the documents included patients' symptoms, habits, medical history, examination results, and diagnostic results. We recruited six undergraduates and one master's student who is a professional in the cardiovascular field to annotate and extract data from the text. After that, we got a structured dataset through the work of data extraction (23). However, the dataset was still missing data. Some features with a missing rate of more than 30% were deleted. Then we used a hierarchical mean filling method to fill in the rest of the data and obtained the final dataset.

In the dataset, each sample contains 62 features, which came from the patients' routine blood examinations, complete biochemical examinations, routine blood coagulation examinations, living habits and family genetic history. Some of these features have been found to be highly correlated with AD, such as D-dimer and Serum potassium (24,25). Through the *t*-test, we chose 42 features with P value less than 0.005 as shown in *Table 2*.

A total of 53,213 patients' data were collected from 2008 to 2016. There were 802 patients diagnosed with AD, the imbalance ratio was about 1:65. In addition, in this dataset, the incidence of AD in men and women was about 2:1; the average age of patients suffering from AD was 56 years old; patients usually had hypotension or hypertension; 33% of the patients suffered from chest pain or abdominal pain.

The prediction performance of predictors

The experimental results are shown in *Tables 3-8* include the average results of ten experiments. In each experiment, the test data set was different, but the size was the same; each included 115 AD patients and 7,487 non-AD patients. After training the predictors, 42 features of each patient in the test set were input into each of the predictors to determine which patients had AD. *Tables 3*-7 show the confusion matrix of the prediction results for AdaBoost (26), XGBoost, SmoteBagging (27), EasyEnsemble (28) and XGBF.

Comparing *Tables 3*-7, it can be seen that the results obtained by AdaBoost and XGBoost are very close, while the latter three, SmoteBagging, EasyEnsemble and XGBF, are significantly different from the first two. For example, in *Table 2*, AdaBoost found 18 AD patients successfully, but 97 AD patients were predicted as non-AD patients; 15 non-AD patients were predicted as AD patients, and 7,472 non-AD patients were predicted correctly. The first two had high accuracy in determining a non-patient was a non-patient. But for patients, their performance was poor and they failed to achieve the purpose of screening. The latter three correctly identified more AD patients. Although they also predicted more non-AD patients as AD patients, the false positive rate was still low considering the large number of negative collections. Such classifiers are obviously more meaningful in disease screening. They greatly reduced the missed diagnosis rate. In the latter three classifiers, XGBF had the best results with the maximum number of correctly predicted AD patients as AD patients. *Table 8* shows the different evaluation results for each algorithm.

The effectiveness of the improved AD screening algorithm XGBF is visualized in *Tables 2*-7. Compared with the traditional ensemble methods of AdaBoost and XGBoost, our method greatly improved sensitivity. AdaBoost and XGBoost cannot deal with imbalanced data. Specificity was higher than 99%, but sensitivity was only about 15% or 16%, which means these algorithms tended to classify all data as non-AD patients. In other words, they did not classify these data at all. Considering two imbalanced data classification methods—SmoteBagging and EasyEnsemble—although the specificity of our results is not obviously dominant, the sensitivity was still higher. In fact, the sensitivity and specificity of XGBF were the highest of all the algorithms. The SmoteBagging model adds extra training data, and the training time became very long. Considering time and sensitivity, EasyEnsemble was better than SmoteBagging. XGBoost was the fastest model, but the sensitivity was poor. AdaBoost was similar to XGBoost. Although the time consumption of XGBF was not the shortest, it was acceptable. Considering all the factors, it achieved the best results.

Discussion

The aim of this study was to develop a machine learning model to screen for early AD from routine medical examination data. Currently, the application of machine learning technology in the medical field has received substantial attention. Wu *et al.* (20) investigated the risk of in-hospital rupture in type A AD patients with a Random

Table 2 *t*-test statistical table of features

Features	AD patient(N=802)	Non-AD patient (N=52,411)	χ^2/t	P
Age	55.57±12.90	62.56±13.06	15.03	<0.001
Sex	574 (71.57%)	29,994 (57.23%)	66.47	<0.001
Chest pain	206 (25.79%)	9,460 (18.05%)	30.99	<0.001
Stomachache	66 (8.23%)	2,996 (5.72%)	9.2	0.002
Heart disease	63 (7.86%)	6,106 (11.65%)	11.1	0.001
Dizziness and headache	62 (7.73%)	7,803 (14.89%)	32.13	<0.001
Aortic valve area murmur	23 (2.87%)	377 (0.72%)	48.88	<0.001
Family history of hypertension	92 (11.47%)	4,798 (9.15%)	5.08	0.024
Chest trauma history	11 (1.37%)	206 (0.39%)	18.62	<0.001
Smoking and duration	10.22±14.39	7.34±13.88	–5.63	<0.001
Hypertension	530 (66.08%)	31,571 (60.24%)	11.29	0.001
Hypertension and duration	6.01±6.47	6.10±7.09	0.36	0.72
Diabetes	88 (10.97%)	11,910 (22.72%)	62.47	<0.001
Diabetes and duration	0.85±2.87	1.82±3.83	9.4	<0.001
Heart rate	81.74±13.87	78.73±14.20	–6.1	<0.001
Systolic pressure	142.41±26.71	136.86±21.90	–5.85	<0.001
Diastolic pressure	83.20±16.59	80.46±13.01	–4.66	<0.001
HGB	119.76±21.57	119.95±22.47	0.24	0.814
NEUT	7.16±4.08	4.79±3.47	–16.35	<0.001
NEUT%	72.83±10.79	65.30±12.09	–19.59	<0.001
LYMPH%	16.94±9.10	24.22±10.44	22.43	<0.001
LYMPH	1.36±0.60	1.57±2.03	9.07	<0.001
MCV	91.84±6.82	92.10±7.17	1.09	0.275
MPV	8.93±1.39	9.36±1.58	8.6	<0.001
TP	64.58±7.06	65.43±8.04	3.41	0.001
ALB	37.08±5.67	38.61±6.26	7.6	<0.001
GLO	27.57±5.19	26.94±5.32	–3.39	0.001
A/G	1.40±0.36	1.49±0.37	6.77	<0.001
TBIL	16.19±21.62	13.20±26.81	–3.86	<0.001
DBIL	6.65±11.52	5.39±13.53	–3.07	0.002
TBA	6.22±13.32	7.55±15.04	2.49	0.013
ALT	66.50±296.27	32.47±108.73	–3.25	0.001
AST	85.34±510.27	36.33±155.39	–2.72	0.007
CRE	136.87±156.07	138.98±213.75	0.28	0.781

Table 2 (*continued*)

Table 2 (*continued*)

Features	AD patient(N=802)	Non-AD patient (N=52,411)	χ^2/t	P
GSP	2.25±0.62	2.03±0.73	−9.79	<0.001
CHO	4.33±0.43	4.37±0.55	2.19	0.029
HDL	1.12±0.17	1.12±0.17	0.28	0.782
LDL	2.60±0.35	2.63±0.46	1.98	0.048
LDH	322.03±684.10	236.51±283.48	−3.54	<0.001
CK	538.04±5,272.64	162.57±567.45	−2.02	0.044
CKMB	35.93±299.32	19.33±33.08	−1.57	0.117
MB	72.69±84.95	57.60±59.02	−5.01	<0.001
K	3.83±0.56	3.97±0.52	7.52	<0.001
Na	139.37±4.28	140.71±3.79	8.77	<0.001
Cl	101.08±4.95	102.59±4.62	8.56	<0.001
CO_2	23.14±3.21	23.20±3.65	0.51	0.609
AG	15.23±3.68	14.95±3.35	−2.38	0.017
Ca	2.16±0.16	2.21±0.18	8.88	<0.001
P	1.19±0.39	1.19±0.34	0.55	0.581
Mg	0.90±0.13	0.89±0.13	−2.38	0.017
ESR	34.94±10.81	35.40±13.92	1.17	0.241
PT%	99.83±18.59	106.62±17.36	10.28	<0.001
INR	1.06±0.39	1.01±0.28	−3.91	<0.001
APTT	37.66±11.29	35.54±9.68	−5.29	<0.001
FIB	4.44±1.81	3.77±1.22	−10.45	<0.001
D-Dimer	1.37±1.94	0.97±1.27	−5.49	<0.001
PLG	252.01±24.57	255.86±27.68	4.4	<0.001
TT	18.92±14.17	19.11±12.91	0.4	0.691
PT	13.57±4.39	13.02±3.06	−3.58	<0.001
ATAG	271.19±17.23	271.18±21.52	−0.01	0.99
FT3	3.91±0.44	3.96±1.15	3.33	0.001
TSH	3.35±2.51	3.41±3.74	0.672	0.502

NEUT, neutrophils; LYMPH, lymphocytes; MCV, mean corpuscular volume; MPV, mean platelet volume; TP, total protein; ALB, albumin; GLO, globulin; A/G, ALB/GLO ratio; TBIL, total bilirubin; DBIL, direct bilirubin; TBA, total bile acid; ALT, alanine aminotransferase; AST, aspartate aminotransferase; CRE, creatinine; GSP, glycosylated serum protein; CHO, cholesterol; HDL, high-density lipoprotein; LDL, low-density lipoprotein; LDH, lactate dehydrogenase; CK, creatine kinase; ESR, erythrocyte sedimentation rate; PT, prothrombin time; INR, international normalised ratio; APTT, activated partial thromboplastin time; FIB, fibrinogen; TT, thrombin time; PT, prothrombin time; ATAG, antithrombin III antigen; FT3, free triiodothyronine-T3; TSH, thyroid-stimulating hormone.

Table 3 Confusion matrix for seven-fold cross validation using AdaBoost[a]

	Predicted positive class	Predicted negative class
Actual positive class	18	97
Actual negative class	15	7,472

[a], in AdaBoost, the number of iterations is 400.

Table 4 Confusion matrix for seven-fold cross validation using XGBoost[a]

	Predicted positive class	Predicted negative class
Actual positive class	18	97
Actual negative class	6	7,481

[a], in XGBoost, the parameter of depth is 4.

Table 5 Confusion matrix for seven-fold cross validation using SmoteBagging[a]

	Predicted positive class	Predicted negative class
Actual positive class	89	26
Actual negative class	1,557	5,930

[a], in SmoteBagging, the number of base classifiers is 100.

Table 6 Confusion matrix for seven-fold cross validation using EasyEnsemble[a]

	Predicted positive class	Predicted negative class
Actual positive class	89	26
Actual negative class	1,550	5,937

[a], in EasyEnsemble, the number of base classifiers is 40.

Table 7 Confusion matrix for seven-fold cross validation using XGBF[a]

	Predicted positive class	Predicted negative class
Actual positive class	92	23
Actual negative class	1,535	5,952

[a], in XGBF, the number of XGboost is 40; m is 1.5; t is 2; n is 2, and k is 5.

Table 8 Seven-fold cross-validation average of five comparison methods

	Sensitivity	Specificity	Time (s)
AdaBoost	16.1%	99.8%	9
XGBoost	15.7%	99.9%	1
SmoteBagging	78.0%	79.2%	1,873
EasyEnsemble	77.8%	79.3%	98
XGBF	80.5%	79.5%	117

Forest model. They used 16 features, including some features extracted from CT images. But there have been no studies specifically developing a screening model from routine examination data. In our study, patients' routine blood tests, biochemical tests and routine tests for blood coagulation were chosen as the candidate features; all of these are basic inspections and can be performed in any hospital, including most rural hospitals with weak facilities. The cost of conducting these inspections is relatively low, and the inspection time is relatively short. At the same time, according to the doctor's experience, some patients' living habits, family history of genetic diseases and other data are also selected. We have used these features to build a machine learning model to predict patients' medical condition. It can achieve higher sensitivity and will help people detect AD in a basic, cheap, and fast way.

In our study, an ensemble learning model XGBF was proposed to get better prediction results. Compared with AdaBoost, XGBoost, SmoteBagging and EasyEnsemble, XGBF combined undersampling, oversampling and the ensemble method to obtain the best results. The average sensitivity of the XGBF algorithm was 80.5%, and the specificity was 79.5%. The results show that the misdiagnosis rate of the XGBF algorithm is lower than that of the other four algorithms. At the same time, the screening results of XGBF were also better than the best results obtained by using Smotebagging in the literature (19), and also better than the clinical misdiagnosis rate (29,30). In particular, the improved algorithm XGBF made the missed diagnosis rate less than 20%, which is less than the missed diagnosis rate of 21.9% (19), 35.5% (29) and 39.69% (30).

This study has some limitations: (I) This study is a retrospective study, so there may be some biases. (II) There are some missing values in the data set. We filled and preprocessed the data manually, so there may be some biases. (III) The parameters of the predictor variables affect the prediction results, but most of the parameters in the experiment were adjusted based on experience or experiment. Therefore, due to the limitation of the number of experiments, this result is the best result we have obtained so far, there may be better results in future.

Conclusions

This study has proposed a machine learning model XGBF to predict the condition of AD with routine medical examination data. This predictor has better prediction effect on imbalanced AD data set than other ensemble algorithms. Therefore, XGBF has practical application value for screening for AD.

Acknowledgments

Funding: We gratefully acknowledge the funding supports from the National Science and Technology Major Project Foundation of China (#2013FY110800); Strategic Emerging Industry Technology Research and Major Technology Achievement Transformation Project (2019GK4013).

Footnote

Reporting Checklist: The authors have completed the TRIPOD reporting checklist. Available at http://dx.doi.org/10.21037/atm-20-1475

Data Sharing Statement: Available at http://dx.doi.org/10.21037/atm-20-1475

Conflicts of Interest: All authors have completed the ICMJE uniform disclosure form (available at http://dx.doi.org/10.21037/atm-20-1475). Dr. Liu and Dr. Tan report that they have a patent "An imbalanced data classification method based on mixed sampling and machine learning pending". The other authors have no conflicts of interest to declare.

Ethical Statement: The authors are accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved. The study was conducted in accordance with the Declaration of Helsinki (as revised in 2013). The study was approved by the ethics board of Xiangya Hospital, Central South University (201502042). This study is a retrospective study, all data were desensitized data from hospital's electronic medical records, which did not contain patient identification information, and the consent was waived.

References

1. Golledge J, Eagle KA. Acute aortic dissection. Lancet 2008;372:55-66.
2. Bossone E, LaBounty TM, Eagle KA. Acute aortic syndromes: Diagnosis and management, an update. Eur Heart J 2018;39:739-49d.
3. Baguet JP, Chavanon O, Sessa C, et al. European Society of Hypertension scientific newsletter: hypertension and aortic diseases. J Hypertens 2012;30:440-3.
4. Ince H, Nienaber CA. Management of acute aortic syndromes. Rev Esp Cardiol 2007;60:526-41.
5. Pape LA, Awais M, Woznicki EM, et al. Presentation, diagnosis, and outcomes of acute aortic dissection: 17-year trends from the International Registry of Acute Aortic Dissection. J Am Coll Cardiol 2015;66:350-8.
6. Kurz SD, Falk V, Kempfert J, et al. Insight into the incidence of acute aortic dissection in the German region of Berlin and Brandenburg. Int J Cardiol 2017;241:326-9.
7. Hagan PG, Nienaber CA, Isselbacher EM, et al. The International Registry of Acute Aortic Dissection (IRAD): new insights into an old disease. JAMA 2000;283:897-903.
8. De León Ayala IA, Chen YF. Acute aortic dissection: an update. Kaohsiung J Med Sci 2012;28:299-305.
9. Kurabayashi M, Miwa N, Ueshima D, et al. Factors leading to failure to diagnose acute aortic dissection in the emergency room. J Cardiol 2011;58:287-93.
10. Hansen MS, Nogareda GJ, Hutchison SJ. Frequency of and inappropriate treatment of misdiagnosis of acute aortic dissection. Am J Cardiol 2007;99:852-6.
11. Asouhidou I, Asteri T. Acute aortic dissection: be aware of misdiagnosis. BMC Res Notes 2009;2:25.

12. Wen W, Zhang XC. A Study on Misdiagnosis Literature of Single Disease of Chinese Misdiagnosed Disease Database: Aortic Dissection. Clinical Misdiagnosis and Mistherapy 2015;28:1-4.
13. Chenkin J. Diagnosis of Aortic Dissection Presenting as ST-Elevation Myocardial Infarction using Point-Of-Care Ultrasound. J Emerg Med 2017;53:880-4.
14. Li Y, Yang N, Duan W, et al. Acute aortic dissection in China. Am J Cardiol 2012;110:1056-61.
15. Dwivedi AK. Performance evaluation of different machine learning techniques for prediction of heart disease. Neural Comput Appl 2018;29:685-93.
16. Gatuha G, Jiang T. Evaluating Diagnostic Performance of Machine Learning Algorithms on Breast Cancer. International Conference on Intelligent Science and Big Data Engineering, 2015:258-66.
17. Liu L, Jiao Y, Li X, et al. Machine learning algorithms to predict early pregnancy loss after in vitro fertilization-embryo transfer with fetal heart rate as a strong predictor. Comput Methods Programs Biomed 2020;196:105624.
18. Huo D, Kou B, Zhou Z, et al. A machine learning model to classify aortic dissection patients in the early diagnosis phase. Sci Rep 2019;9:2701.
19. Liu L, Zhang C, Zhang G, et al. A study of aortic dissection screening method based on multiple machine learning models. J Thorac Dis 2020;12:605-14.
20. Wu J, Qiu J, Xie E, et al. Predicting in-hospital rupture of type A aortic dissection using Random Forest. J Thorac Dis 2019;11:4634-46.
21. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016:785-94.
22. Chawla NV, Bowyer KW, Hall LO, et al. SMOTE: Synthetic Minority Over-sampling Technique. J Artif Intell Res 2011;16:321-57.
23. Gao Y, Wang Y, Wang P, et al. Medical Named Entity Extraction from Chinese Resident Admit Notes Using Character and Word Attention-Enhanced Neural Network. Int J Environ Res Public Health 2020;17:1614.
24. Marill KA. Serum D-dimer is a sensitive test for the detection of acute aortic dissection: a pooled meta-analysis. J Emerg Med 2008;34:367-76.
25. Chen Z, Huang B, Lu H, et al. The effect of admission serum potassium levels on in-hospital and long-term mortality in type A acute aortic dissection. Clin Biochem 2017;50:843-50.
26. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55:119-39.
27. Ali A, Shamsuddin SM, Ralescu A. Classification with class imbalance problem: a review. Int J Adv Soft Comput 2015;7:176-204.
28. Liu XY, Wu J, Zhou ZH. Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cybern B Cybern 2009;39:539-50.
29. Chen XF, LI XM, Chen XB, et al. Analysis of Emergency Misdiagnosis of 22 Cases of Aortic Dissection. Clinical Misdiagnosis and Mistherapy 2016;29:30-1.
30. Teng Y, Gao Y, Feng S, et al. Analysis of Emergency Misdiagnosis of 131 Cases of Aortic Dissection. Chinese Journal of Misdiagnostics 2012;8:1873.

Cite this article as: Liu L, Tan S, Li Y, Luo J, Zhang W, Li S. An early aortic dissection screening model and applied research based on ensemble learning. Ann Transl Med 2020;8(23):1578. doi: 10.21037/atm-20-1475

Supplementary

The following is a brief introduction to the algorithms we used.

1. AdaBoost

AdaBoost, which is called Adaptive Boosting, is a machine learning algorithm based on the boosting idea. AdaBoost is an iterative algorithm. The core idea is to use different learning algorithms for the same training set, train different weak classifiers, and then combine these weak classifiers to construct a final strong classifier.

Most Boosting methods change the distribution of data by changing the weight of the training data so that it can learn a series of different weak classifiers. In AdaBoost, there are two weights. The first is that each sample in the training set has a weight. For each sample that is misclassified by the weak classifiers in the last round of training, the weight is increased and the weight of the correct classification sample is reduced. Thus, in the next round of training, the misclassified sample is highly attended. In the second, each weak classifier has a weight. For a weak classifier with a relatively small classification error rate, the weight is increased, and the weight of the weak classifier which has a large classification error rate is reduced. Thus, the accuracy of the strong classifier in improved when the algorithm is finally integrated.

2. XGBoost

XGBoost is a gradient boosting decision tree. Its full name is Extreme Gradient Boosting, which is an extension of Gradient Boosting. The Boosting classifier is an ensemble learning model, whose basic idea is to combine hundreds of tree models that have lower classification accuracy into a model with high accuracy. The model is continuously iterated and generates a new tree for each iteration. Determining how to generate a reasonable tree at each step is the core of the Boosting classifier. The Gradient Boosting algorithm uses the idea of gradient descent in generating each tree. Then based on all the trees generated in the previous step, it moves in the direction of minimizing the given objective function. Under reasonable parameter settings, a certain number of trees need to be generated to achieve the expected accuracy. When the data set is large and complex, the Gradient Boosting algorithm has a huge amount of computation. XGBoost is an implementation of Gradient Boosting that automatically uses multithreading of the CPU for parallel operations and it could improve accuracy by improving the algorithm.

XGBoost's base learners are regression trees. Its loss function uses second-order Taylor expansion. It has high accuracy, and is not easy to overfitting, and is scalable. It can process high-dimensional sparse features distributedly. Therefore, Xgboost is 10 times faster than similar algorithms under the same circumstances.

3. SmoteBagging

SmoteBagging is an ensemble learning algorithm that uses voting strategies. As you can see from the name, the SmoteBagging algorithm is a method that uses the Smote and Bagging methods. Smote is a method that could artificially synthesize new samples, while Bagging samples the training set in a way that sampling with replacement to construct different training sets for each base classifier, and it usually adopts a simple voting method for decision output. When creating each base classifier for voting, the Nn majority sample will be sampled first, and then the same number of Nn minority samples will be constructed. The minority samples are obtained by sampling with replacement and smoting, and the proportion is determined according to the percentage b%, where b% is a multiple of 10% between 10% and 100%. That is, in each base classifier, a few samples are sampled by sampling with replacement whose ratio is Nn*b%, while the proportion of the artificially synthesized samples using Smote is Nn*(1-b%).

In each iteration, SmoteBagging can choose the method for multiplying the number of samples of majority class. In this process, the minority samples with an insufficient number are generated by smote algorithm, and selecting different numbers of majority samples in each iteration improves the difference of base classifiers. SmoteBagging is an algorithm that uses the idea of oversampling, but it does not simply use one of the methods of Smote or Bagging. It not only reduces the overfitting that the Bagging method may produce, but also reduces the negative impact of the sample of the artificial synthesis in the Smote method. In addition, since b% is a multiple of 10% between 10% and 100%, the diversity of base classifiers is guaranteed. Diversity is very important in the improvement of classification accuracy and generalization of the model.

4.EasyEnsemble

EasyEnsemble is a common algorithm that uses

http://dx.doi.org/10.21037/atm-20-1475

undersampling to deal with the imbalance problem. Unlike direct undersampling, it undersamples multiple balanced data sets and uses these data sets to train multiple classifiers. Finally, these classifiers are combined by some strategies.

EasyEnsemble can be described as follows. Suppose that the minority of the training data set is P, the majority is N, and $|N| >> |P|$ is satisfied. The data sets of the majority are divided into N sub-data sets of N1, N2, N3...NT, and satisfy $|Ni| = |P|$. For each data set Ni, we combine it with P as a training set to train a classifier Hi, in which AdaBoost is used to train the classifier Hi. Finally, T classifiers are obtained and combined to form the final model.

The idea behind EasyEnsemble is very simple. In order to avoid data imbalance, the algorithm samples T-balanced sub-data sets. Each of them includes all minority and partial majority of the initial samples. In addition, a simple average method, rather than a voting method, is used in constructing the final classifier so that we can obtain information from all samples. Besides, EasyEnsemble is an ensemble algorithm based on AdaBoost, and because the AdaBoost algorithm is also an ensemble algorithm, the EasyEnsesmble algorithm is also called ensemble of ensemble algorithm.

5.Extreme Gradient Boosting forest(XGBF)

The EasyEnsesmble algorithm does not generate new data, so it learns quickly. The information provided by each sample is completely obtained by using it. However, since there are too few samples of majority class in each subset, the subclassifier cannot obtain more majority information. In addition, due to the objective distribution of data and the objective conditions of data collection, different types of samples tend to have similar values on certain features. Because these samples are located in overlapping areas of the feature space, they are called overlapping samples, and the problems caused by them are also called class overlapping problems, which may result in poor classification. The EasyEnsesmble algorithm is not ideal for this problem.

AdaBoost and XGBoost are currently better algorithms for processing general data. They are not targeted on large-scale unbalanced data sets, so it is very likely that their performance will be poor.

SmoteBagging consumes a lot of time due to its excessive smote operation and may affect the true distribution of data.

Therefore, in view of the shortcomings of the above algorithms, our study proposes an oversampling ensemble algorithm named Extreme Gradient Boosting Forest (XGBF). The algorithm combines the strong points of undersampling, oversampling and ensemble. At the same time, it also merges with XGBoost because it has the advantage of flexibility, high precision, short training time, and prevention of over-fitting.

XGBF, as an ensemble learning algorithm, is composed with several XGBoost classifiers. XGBoost itself is a boosting ensemble model using decision trees as its base classifier. The specific structure of the XGBF model is shown in Figure S1.

Since XGBoost has a fast convergence rate, the operation speed of XGBF can be guaranteed. The training data entered into each XGBoost classifier are composed with some undersampled majority data and oversampled minority data. In the XGBF algorithm, the oversampling operations are performed on the minority class, which includes duplicating and smote. The minority samples are strengthened by these two methods, so the learning model can get more information from the minority samples. The undersampling operation is performed on the majority class so that the distribution of each kind of sample is balanced. Finally, each weak subclassifier is enhanced by ensemble methods to achieve better classificion results.

Algorithm 1 XGBF Algorithm

```
1. $i$=0
2. Duplicate all minority samples in $P$ $t$ times to get
   duplicate dataset $DP$
3. For each sample $x$ in $DP$, repeat somte operation $n$
   times to get dataset $P'$
4. While $i$<= $tb$ repeat
   4.1 $i$=$i$+1
   4.2 Sampling $m * |P|$ samples into $N_i$ from $N$ without
       replacement
   4.3 Using $N_i$ and $P'$ train an XGBoost classifier $H_i$
5. Output a weighted average of the results of tb
   classifier $H$
```

The XGBF algorithm combines the advantages of the oversampling, undersampling and ensemble methods so that the classifier can fully learn the characteristics of the samples and produce better results. The pseudo code for the XGBF algorithm is shown in algorithm 1. In the algorithm, P represents the minority class dataset, which is the patients' set; N represents the majority class dataset, which is the non-patient's set; and tb represents the number of XGBoost classifiers. |P| and |N| mean the cardinality of the set P and N, respectively

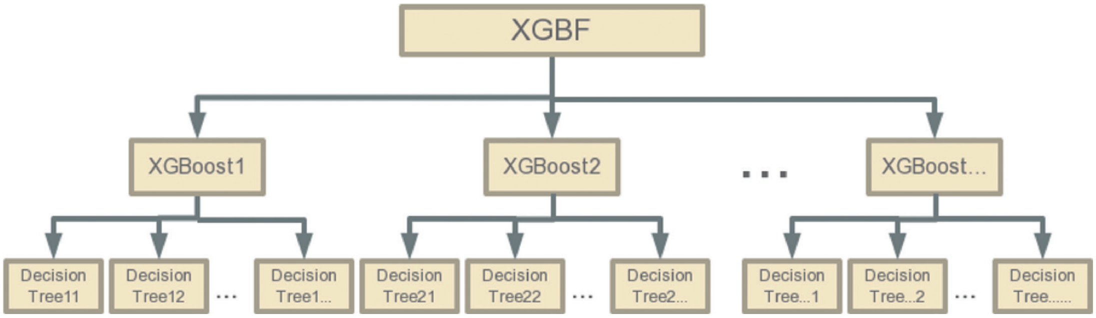

Figure S1 The specific structure of the XGBF model.

http://dx.doi.org/10.21037/atm-20-1475