

Prediction of Cerebral Aneurysm Rupture

Qiangfeng Peter Lau¹ Wynne Hsu¹ Mong Li Lee¹ Ying Mao² Liang Chen²

¹School of Computing
National University of Singapore
{plau,whsu,leeml}@comp.nus.edu.sg

²Neurosurgical Department
Huashan Hospital, China
yingmaoc@online.sh.cn, clclcl95@shmu.edu.cn

Abstract

Cerebral aneurysms are weak or thin spots on blood vessels in the brain that balloon out. While the majority of aneurysms do not burst, those that do would lead to serious complications including hemorrhagic stroke, permanent nerve damage, or death. Yet, surgical options for treating cerebral aneurysms carry high risk to the patient. It is vital for the doctors to accurately diagnose aneurysms that have high probabilities of rupturing. In this application, the patient dataset has many attributes, ranging from patient profile to results from diagnostic test and features extracted from brain images. Many of the attributes are discrete and have missing values. The dataset is also highly biased, with 15% unrupture cases and 85% rupture cases. Building a classifier that unerringly predicts the unrupture (rare) class is a challenge. In this paper, we describe a systematic approach to build such a classifier through suitable combination of data mining algorithms. Our approach automatically determines the optimal combination of these algorithms for a dataset. The system has an accuracy of 92% and is currently being deployed at the Huashan Hospital.

1. Introduction

Cerebral aneurysm are common in adult populations, however the majority of them are small and do not rupture during a person's lifetime [5]. Advances in imaging technology have led to more aneurysms being detected in an unruptured state, resulting in a heavier emphasis on the correct treatment of unruptured aneurysms [19]. Studies on rupture risk using standard statistical techniques have been carried out to identify individual risk factors [12, 11]. However, the disease's natural history, and reasons for rupture remain largely unknown [5]. Hence the treatment of unruptured aneurysms is still a controversial topic [8]. One important consideration in determining an appropriate treat-

ment is whether an aneurysm has a high probability of rupture. Surgery for unruptured aneurysms carries high risk. Yet, failure to perform surgery may result in death when the aneurysm ruptures. Hence, it is critical to be able to accurately predict aneurysm ruptures.

Existing approaches to building accurate predictors range from decision trees [15, 7], to support vector machines [14], to Naïve Bayes classifiers [17]. Many techniques have also been investigated to handle special data characteristics such as class imbalance [1, 18], irrelevance and redundancy [6, 23], and unequal costs of misclassifications. However, their accuracy when used in combination is not well established. These combinations are further compounded by the bias experienced by different classification algorithms on different datasets.

Our solution is to design a methodology that systematically determines the best combination of algorithms, drawing these from sets of data-mining algorithms that perform different tasks. The best combination is then used to build a model for a prediction tool. Through a large-scale experiment using our proposed methodology evaluated on a real world hospital dataset and 12 UCI datasets [2], we make the following contributions:

1. We describe the challenges faced in this domain that are echoed in other domains and report the usefulness of our methodology.
2. We show that strictly combining the best algorithms of individual tasks often do not result in the optimal combination. Further, for multiple ensembles, it is difficult to establish the number of base algorithms to combine.
3. We implement an aneurysm rupture prediction tool using our experiment results. However, even with an accurate system, the justification of predictions is vital to the user. Thus, we propose an intuitive justification scheme for a novice user.

2. Issues in Cerebral Aneurysm Rupture Prediction

In this section, we discuss the three issues in devising a predictor for cerebral aneurysm rupture. These issues are also present in other prediction domains.

2.1. Flexible Class Definition & Labeling

In cerebral aneurysm rupture prediction, it is important to allow for flexible class definition. This allows doctors to have greater flexibility in adapting the prediction system to suit current medical practices. The class label of each patient record is automatically computed from a given definition. Here, we construct the class label based on a chosen number of months, n , a patient has an unruptured aneurysm. We choose n to correspond to the current regular consultation interval of patients at the hospital.

We use L_{NR} to denote the class “will not rupture”. This class refers to patients who have no prior surgery and: (1) have not undergone surgery and are alive, or (2) have undergone surgery more than n months after the detection of the disease – indicating that the aneurysm remains unruptured for at least n months. The class L_R corresponds to “will rupture” and it refers to patients that have prior ruptures before surgery can be performed, or have died without treatment. In our experiments for the aneurysm dataset, we set $n = 12$.

2.2. Unequal Misclassification Costs

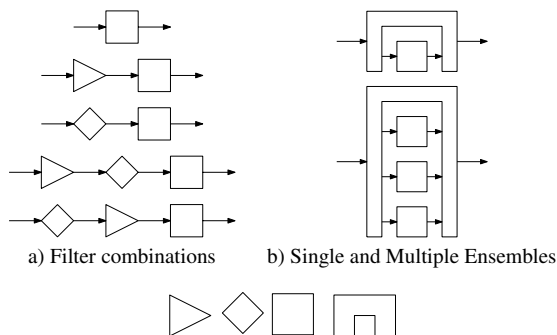
In designing a predictor for cerebral aneurysm rupture, the uneven cost between false predictions is an important factor. Predicting a false will rupture case is less costly than predicting a false will not rupture case. Patients suffering from a ruptured aneurysm have a high mortality rate and survivors have a high morbidity rate. However, surgical treatment of unruptured aneurysms have mortality and morbidity rates that although high, are much lower than the ruptured case [5]. Patients will undergo surgical treatment if doctors believe that their aneurysm is likely to rupture. For an inaccurate prediction, there are two possible scenarios:

1. A will rupture case is misclassified. There is a high chance the patient will die or be paralyzed.
2. A will not rupture case is misclassified. This will result in a patient needlessly suffering a traumatic brain surgery.

From the above, it is evident that it is more important to minimize scenario 1 than its counterpart. Hence there should be a high *precision* for class L_{NR} .

2.3. Class Imbalance

Another factor that influences the design of the cerebral aneurysm rupture predictor is the rare class problem. Only a minority of patients with aneurysms will suffer from an aneurysm rupture and there is a tendency to focus on more serious cases of the disease. Unless long term efforts are in place for record keeping of minor cases, a typical real world dataset from a hospital is skewed towards ruptured cases as the majority in contrast to the population distribution. As a result, many medical datasets are imbalanced with L_R being the majority.



Legend, algorithm types (left – right): Feature Selection/Transformation, Class Imbalance, Classification, Ensemble

Figure 1. Algorithm Combinations

3. Methodology

Previous research has established many algorithms to handle different aspects of data mining issues. Studies have looked into filtering techniques for handling class imbalance problems [1, 18], feature selection techniques to reduce the high dimensionality problem in datasets [6, 23], classification techniques, and ensemble techniques for building good predictors [4, 7, 3]. However, real world applications require identifying the best combination of these algorithms such that they can produce the optimal predictor [22, 13]. A multitude of variables are introduced when determining the best combination of algorithms to employ and the technique of determining the best is not well established. A combination of algorithms may include preprocessing using filters (Figure 1a), ensemble techniques such as Bagging, or more complicated multiple classifier ensemble methods like Stacking [3] (Figure 1b). Also, there may be a need to change the class definition in accordance with medical practice giving rise to a new dataset. Performance of a combination of algorithms is dependent on the dataset, thus the best combination must be re-evaluated. These considerations justify the need for a structured reusable approach towards deriving the best combination from a finite subset of combinations.

The life-threatening risk of an inaccurate prediction, commonly associated with medical domains, calls for rigorous evaluation of algorithm combinations. The methodology we present is limited only by the computational time required to evaluate larger datasets – a problem manageable through distributed computing and the selection of more efficient algorithms.

The methodology given in Algorithm 1 depicts a framework to limit the number of algorithm combinations by specifying input sets of base algorithms for the tasks: classification (\mathbf{C}_{base}), feature selection/transformation (\mathbf{F}_{fea}), class imbalance (\mathbf{F}_{imb}), single ensembles (\mathbf{E}_1), and multiple ensembles (\mathbf{E}_n). For simplicity, we define the binary function,

$$\diamond(\mathbf{G}_1, \mathbf{G}_2) = \{g_2 \circ g_1 | g_1 \in \mathbf{G}_1 \wedge g_2 \in \mathbf{G}_2\}$$

where \circ is the function composition operator, \mathbf{G}_1 and \mathbf{G}_2 are sets of functions. Like \circ , \diamond is asymmetric. Note that \diamond is a binary operator and the notation $\mathbf{G}_1 \diamond \mathbf{G}_2$ is equivalent to $\diamond(\mathbf{G}_1, \mathbf{G}_2)$.

Algorithm 1 Find Best Combination

```

1: Input:  $d, \mathbf{C}_{base}, \mathbf{F}_{fea}, \mathbf{F}_{imb}, \mathbf{E}_1, \mathbf{E}_n, \sigma, \mu$ 
2: Output:  $c_{best}$  – the best combination
3:
4: /* Stage 1 */
5:  $\mathbf{F}_{cmb} \leftarrow \mathbf{F}_{fea} \cup \mathbf{F}_{imb} \cup (\mathbf{F}_{fea} \diamond \mathbf{F}_{imb}) \cup (\mathbf{F}_{imb} \diamond \mathbf{F}_{fea})$ 
6:  $\mathbf{C}_{cmb} \leftarrow \mathbf{C}_{base} \cup (\mathbf{F}_{cmb} \diamond \mathbf{C}_{base})$ 
7:
8: /* Evaluate runs 10-fold CV on  $d$  */
9: EVALUATE( $\mathbf{C}_{cmb}, d$ )
10:
11: /* Stage 2, Single Ensembles */
12:  $\mathbf{C}_\sigma \leftarrow \{c | c \in \mathbf{C}_{cmb} \wedge rank(c) \leq \sigma\}$ 
13:  $\mathbf{S} \leftarrow \{\mathbf{C} | \mathbf{C} \in 2^{\mathbf{C}_\sigma} \wedge \mathbf{C} = 1\}$ 
14:  $\mathbf{C}_{se} \leftarrow \{e(\mathbf{C}) | e \in \mathbf{E}_1 \wedge \mathbf{C} \in \mathbf{S}\}$ 
15:
16: /* Multiple Ensembles */
17:  $\mathbf{C}_\mu \leftarrow \{c | c \in \mathbf{C}_{cmb} \wedge rank(c) \leq \mu\}$ 
18:  $\mathbf{N} \leftarrow \{\mathbf{C} | \mathbf{C} \in 2^{\mathbf{C}_\mu} \wedge \mathbf{C} \geq 2\}$ 
19:  $\mathbf{C}_{me} \leftarrow \{e(\mathbf{C}) | e \in \mathbf{E}_n \wedge \mathbf{C} \in \mathbf{N}\}$ 
20:
21: EVALUATE( $\mathbf{C}_{se} \cup \mathbf{C}_{me}, d$ )
22:  $\mathbf{C}_{total} \leftarrow \mathbf{C}_{cmb} \cup \mathbf{C}_{se} \cup \mathbf{C}_{me}$ 
23:
24: /* output the best */
25: output BEST( $\mathbf{C}_{total}$ )

```

The methodology comprises two Stages. In each stage, combinations of these base algorithms are generated in a structured manner. Each combination of algorithms is evaluated on the dataset, d . After Stage 1, the number of combinations to be evaluated in Stage 2 will be pruned by input parameters, $\sigma \in \mathbb{Z}^+$ and $\mu \in \mathbb{Z}^+$. The details of the algorithm and its inputs are described in the following sections.

3.1. Evaluation

Selecting the best combination of algorithms require a metric for evaluation of prediction quality. By selecting a metric that reflects the uneven costs of inaccurate predictions, the methodology can internalize these costs and reward combinations accordingly.

The cost of predicting a false \mathbf{L}_{NR} is higher than predicting a false \mathbf{L}_R . The *Precision* for the class \mathbf{L}_{NR} is calculated as $\frac{T(\mathbf{L}_{NR})}{T(\mathbf{L}_{NR})+F(\mathbf{L}_{NR})}$ and the *Recall* as $\frac{T(\mathbf{L}_{NR})}{T(\mathbf{L}_{NR})+F(\mathbf{L}_R)}$, where $T(X)$ and $F(X)$ denotes respectively, the number of true or false predictions for the class label, X . To include both *Precision* and *Recall* in the evaluation metric, we use the weighted *F-measure* [16] given as,

$$F_\beta = \frac{(1 + \beta) \times Precision \times Recall}{\beta \times Precision + Recall}$$

Since it is more important to predict \mathbf{L}_{NR} correctly compared to \mathbf{L}_R , we use $F_{0.5}$ as our evaluation metric, i.e. the *F-measure* of \mathbf{L}_{NR} with a heavier weight for *Precision*. To evaluate an algorithm, 10-fold cross validation is performed on the dataset and $F_{0.5}$ is calculated.

To place scoring emphasis on class \mathbf{L}_{NR} may seem contrary to the convention for predictions in medical domains. However, in our case, this is justifiable as \mathbf{L}_{NR} is the minority class, resulting in our chosen metric being more sensitive to misclassifications of \mathbf{L}_{NR} instances.

3.2. Base Classifiers

We use \mathbf{D} to represent the set of all datasets, and \mathbf{M} to denote the set of all classification models. A classification algorithm, $c : \mathbf{D} \mapsto \mathbf{M}$, takes a dataset as input and outputs a classification model. Suppose \mathbf{C}_{all} is the infinite set of all classification algorithms, we select a starting set of base classifier algorithms, $\mathbf{C}_{base} \subset \mathbf{C}_{all}$, as one of the inputs for Algorithm 1.

For our experiments, we selected six classification algorithms for \mathbf{C}_{base} . They were chosen to represent different approaches towards classification and are given as follows:

- (1-2) C4.5 for decision trees [15] and a rule based variant.
- (3-4) SMO, a variant of SVM with a linear kernel and quadratic polynomial kernel for functions [14]
- (5) Averaged One-Dependence Estimators (AODE) for Bayes [17]
- (6) 5 nearest neighbors lazy classification (using HVDM [20] to evaluate distance) for instance based learning.

3.3. Filters

A filter algorithm, $f : \mathbf{D} \mapsto \mathbf{D}$ takes an input dataset and produces an output dataset with the same target feature.

Suppose \mathbf{F}_{all} is the set of all filter algorithms, the function $c \circ f \in \mathbf{C}_{all}, f \in \mathbf{F}_{all} \wedge c \in \mathbf{C}_{all}$, is a combination of a filter and classification algorithm, i.e. building a model with a dataset will first filter it with f and then build with c .

Feature selection and transformation algorithms can be used to reduce irrelevant and redundant data, improving classifier accuracy [6]. We use these techniques to evaluate the quality of features from a dataset that has missing values. For our application, feature selection is useful since there is imperfect domain knowledge. We select a starting set of feature selection and transformation algorithms, $\mathbf{F}_{fea} \subset \mathbf{F}_{all}$, as one of the inputs for Algorithm 1.

A total of eight filter algorithms were chosen for \mathbf{F}_{fea} in our experiment. They are:

- (1) FCBF that uses the Symmetrical Uncertainty (SU) metric to analyze relevance and redundancy [23]
- (2-4) Filtering features, by analyzing relevance only using the SU metric, that exceed the threshold values of 0.02, 0.03 and 0.04
- (5-7) Principle Component Analysis (PCA) and select the top 2, 3, and 4 features
- (8) PCA to transform all features back to feature space after the initial transformation to reduce noise.

Most real world datasets have rare class or rare cases [18]. Several filter methods exist for addressing the imbalance problem that can lead to better rare class classification accuracy than without them [1]. These filter algorithms modify datasets by creating or removing records.

Predicting the minority class accurately is of paramount importance. Thus, we choose 17 filter algorithms as $\mathbf{F}_{imb} \subset \mathbf{F}_{all}$ for our experiment. Most of these algorithms either sample or create synthetic instances until a certain minority to majority class ratio. Unless otherwise stated, each algorithm was used in two minority to majority class ratio variants of 1 : 2 and 1 : 1. They are:

- (1-2) Random over-sampling minority class
- (3-4) Random under-sampling majority class
- (5) Tomek Links used to under-sample the majority class without conforming to the ratio
- (6-7) Synthetic Minority Over-sampling TEchnique (SMOTE)
- (8-9) SMOTE followed by Tomek Links. SMOTE was modified to create synthetic instances up to a specified minority to majority ratio.
- (10-11) Cluster-Based Over-sampling
- (12-13) Cluster-Based SMOTE – SMOTE is used to create synthetic instances instead of over-sampling
- (14-15) Cluster-Based Over-sampling followed by Tomek Links
- (16-17) Cluster-Based SMOTE followed by Tomek Links.

Note that algorithms (1) – (9) are selected from those investigated in [1], while algorithms (10) – (17) are selected or modified from [9]. We also modify the cluster-based algorithms to allow inflation to a given minority to majority class ratio.

There is no established convention on the sequence in which the two filter types, $f_{fea} \in \mathbf{F}_{fea}$ and $f_{imb} \in \mathbf{F}_{imb}$, should be applied. Thus they may be combined in two different ways, giving functions $f_{fea} \circ f_{imb}$ and $f_{imb} \circ f_{fea}$ where $f_{fea} \circ f_{imb}, f_{imb} \circ f_{fea} \in \mathbf{F}_{all}$. Hence the combination of filters,

$$\mathbf{F}_{cmb} = \mathbf{F}_{fea} \cup \mathbf{F}_{imb} \cup (\mathbf{F}_{fea} \diamond \mathbf{F}_{imb}) \cup (\mathbf{F}_{imb} \diamond \mathbf{F}_{fea})$$

in Algorithm 1, Line 5. Using \mathbf{F}_{cmb} and \mathbf{C}_{base} , we combine them to give a new set of classification algorithms, $\mathbf{C}_{cmb} = \mathbf{C}_{base} \cup (\mathbf{F}_{cmb} \diamond \mathbf{C}_{base})$, as shown in Algorithm 1, Line 6 and illustrated in Figure 1a.

For Stage 1, each $c \in \mathbf{C}_{cmb}$ is evaluated in Algorithm 1, Line 9 using the method described in Section 3.1 and the algorithms are ranked (in descending order) according to their $F_{0.5}$ values.

3.4. Ensembles

Ensembles are meta-classification algorithms used to enhance base classification algorithms. Single ensembles use one classification algorithm, build different models by varying the dataset (e.g. Bagging) and finally combine the various models. Multiple ensembles use multiple classification algorithms and a strategy (e.g. meta-classifier in Stacking) to combine their predictions. Both ensemble techniques have been shown to perform better than the original base classification algorithm(s) without modifications [7, 3].

For our purposes, ensemble algorithms are not further nested within each other due to the computational time incurred. Also, we introduce two greedy pruning parameters σ and μ to limit the number of ensemble combination evaluations. These parameters are based on the assumption that the best ensemble algorithm will consist of the top σ and μ algorithms from Stage 1 for single and multiple ensembles respectively.

Let \mathbf{E}_{all} be the set of all ensemble algorithms and 2^X denote power set of X . The function, $e : 2^{\mathbf{C}_{all}} \mapsto \mathbf{C}_{all}$, is an ensemble algorithm that maps a set of classification algorithms, $\mathbf{C} \in 2^{\mathbf{C}_{all}}$, to an ensemble classification algorithm, $e(\mathbf{C}) \in \mathbf{C}_{all}$. The resulting algorithm combinations are illustrated schematically in Figure 1b. Note that \mathbf{C} can contain any classifier algorithm illustrated in Figure 1a.

For single ensembles, there is no guarantee that the best combination can be found solely based on evaluating ensembles of the best algorithm from Stage 1 – a base classifier that performs poorer than another might perform better after Bagging is applied to each. However, to reduce the

Dataset	# fea.	size	min.(%)	maj.(%)	miss.(%)	num.(%)	nom.(%)	class (min, max)
ecoli	7	336	10.42	89.58	0.00	100.00	0.00	imU, remainder
glass	9	214	7.94	92.06	0.00	100.00	0.00	vehic_win_float, remainder
flags	29	194	8.76	91.24	0.00	6.90	93.10	white, remainder
aneurysm	77	414	14.98	85.02	14.83	58.44	41.56	L_{NR}, L_R
sponge	45	76	7.89	92.11	0.64	0.00	100.00	AMBOS_BLANDO, DURO ¹
zoo	17	101	9.90	90.10	0.00	5.88	94.12	invertebrate, remainder
vehicle	18	846	23.52	76.48	0.00	100.00	0.00	van, remainder
hepatitis	19	155	20.65	79.35	5.67	31.58	68.42	DIE, LIVE
autos	25	205	13.17	86.83	1.15	60.00	40.00	3, remainder ²
credit-german	20	1000	30.00	70.00	0.00	35.00	65.00	bad, good
haberman	3	306	26.47	73.53	0.00	66.67	33.33	2, 1
wine	13	178	26.97	73.03	0.00	100.00	0.00	3, remainder
primary-tumor	17	339	8.55	91.45	3.90	0.00	100.00	ovary, remainder

¹ 'P.Sustrato' as class feature ² 'symboling' as class feature

Table 1. Datasets characteristics & description

number of single ensemble evaluations, we only consider the top σ algorithms. From Algorithm 1, Lines 12 to 14, the set of classification algorithms from the evaluated set of C_{cmb} to be combined is $C_\sigma = \{c | c \in C_{cmb} \wedge rank(c) \leq \sigma\}$.

In the single ensemble case, only one classification algorithm is combined. Let $S = \{C | C \in 2^{C_\sigma} \wedge \bar{C} = 1\}$ be the set of algorithm sets to combine that each contains a single member. We select a set of single ensemble algorithms, $E_1 \subset E_{all}$, as input to Algorithm 1. Then, the set of single ensemble algorithm combinations to evaluate is $C_{se} = \{e(C) | e \in E_1 \wedge C \in S\}$.

For our experiment, we selected $\sigma = 20$. The following two single ensemble algorithms investigated in [7] are selected for the input E_1 in our experiment:

- (1) Bagging using 10 sampled datasets
- (2) AdaBoost for 10 rounds

For multiple ensemble algorithms, it is difficult to determine the optimal number of models to combine [10]. Using a subset of the algorithms combined may out-perform the full set. However, to reduce the number of evaluations, we only combine subsets from size 2 to μ of the top μ algorithms from the evaluated set of C_{cmb} . From Algorithm 1, Lines 17 to 19, the set of top μ algorithms whose subsets are to be used in multiple ensemble combinations is $C_\mu = \{c | c \in C_{cmb} \wedge rank(c) \leq \mu\}$.

In the multiple ensemble case, two or more classifiers in C_μ may be combined per ensemble. Let $N = \{C | C \in 2^{C_\mu} \wedge \bar{C} \geq 2\}$, be the set of algorithm sets to combine. We select a set of multiple ensemble algorithms, $E_n \subset E_{all}$, as input to Algorithm 1. Then, the set of multiple ensemble algorithm combinations to evaluate is $C_{me} = \{e(C) | e \in E_n \wedge C \in N\}$.

For our experiment, we selected $\mu = 6$. Five multiple ensemble algorithms are selected, namely:

- (1) Voting – unweighted sum of base algorithm predictions

- (2-5) Stacking [3] using AODE, SMO SVM with linear and quadratic polynomial kernels, and C4.5 as meta-classification algorithms

After evaluating the ensemble classifier algorithms given by $C_{se} \cup C_{me}$ (Algorithm 1, Line 21), the set, $C_{total} = C_{cmb} \cup C_{se} \cup C_{me}$, is set of all evaluated classifier algorithms. Ranking C_{total} and taking the top performer will produce the best combination of data mining algorithms for the dataset (Algorithm 1, Line 25).

4. Experiment Results

The evaluation of the combinations of algorithms are done using the algorithms in WEKA [21] augmented with those absent (notably those that address class imbalance issues). We implemented a client-server application to capitalize on distributed computing. The server distributes combinations for clients to evaluate and consolidates the results. Communication between the client and the server is done over TCP/IP.

We carried out experiments on the aneurysm dataset and, 12 UCI datasets [2] with varying minority class rarity to further investigate the effects of our methodology on other domains. A total of more than 50000 models were evaluated. Many of the UCI datasets were modified into binary class problems by merging other classes to create a rare class. Table 1 summarizes the details of the datasets.

4.1. Score Improvement

Figure 2 plots the majority/minority class ratio versus the gain in $F_{0.5}$ score. For each dataset, the gain in score is measured by subtracting the score for the best base classification algorithm from score of the best algorithm combination. We observe that all of the datasets show positive gains and a majority of the datasets gain between 5 to 15 points with the mean gain being 10.2 points.

Figure 3 shows the gain in $F_{0.5}$ scores at post Stage 1 and Stage 2 respectively. The mean gain at post Stage 1 is 7.7 points and mean gain between Stage 1 and Stage 2 is 2.5 points. This shows that most of the improvement made by our methodology is in Stage 1. For very large datasets, where ensemble techniques become infeasible, Stage 2 of the methodology may be discarded. However, for predicting aneurysm ruptures, a wrong prediction may be fatal – any increase in accuracy is desirable. Hence, our methodology is best suited when the benefits of accurate classification far outweigh the cost of the extra computation time incurred.

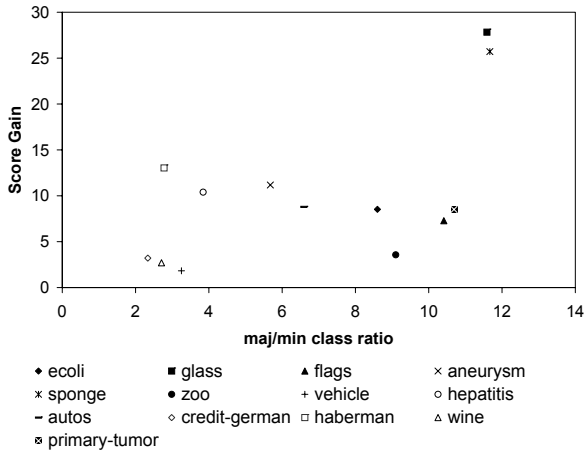


Figure 2. Class ratio vs. $F_{0.5}$ score gain

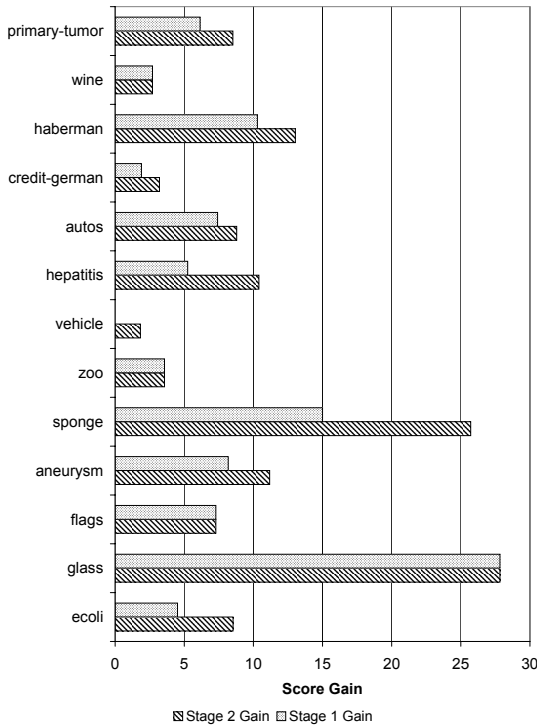


Figure 3. Improvement in $F_{0.5}$ score

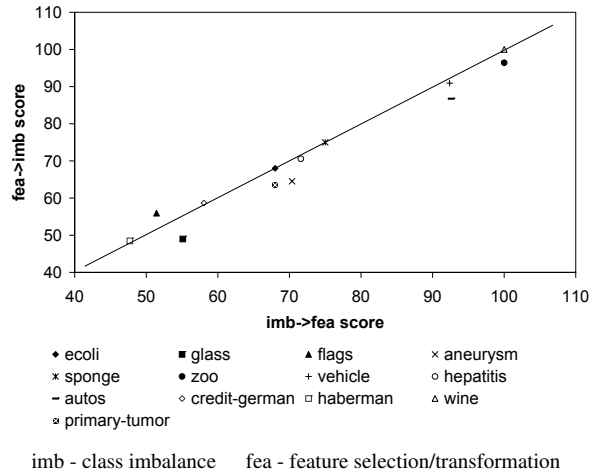


Figure 4. $F_{0.5}$ scores for feature selection and class imbalance combinations

4.2. Filter Combination

Figure 4 shows the $F_{0.5}$ scores for the two sequences in which feature selection and class imbalance algorithms may be applied. Most of the datasets perform better when class imbalance is addressed before feature selection. However, this is not always the case as seen in the *flags* dataset.

Combining the best filter algorithms for each task may not always result in the optimal combination. Using Table 2 as the legend, Table 3 illustrates this for the aneurysm dataset. $SU_{0.02}$ (ranked 7) feature selection out-performs $SU_{0.04}$ (ranked 8) when used alone. Yet, when used in combination with a class imbalance algorithm, combinations that contain $SU_{0.04}$ (ranked 1 and 2) out-perform those with $SU_{0.02}$ (ranked 3 and 4). Furthermore, although over-sampling (ranked 5) may out-perform Cluster Based SMOTE (ranked 6), when used with feature selection, the converse is true. Hence it is justifiable for our methodology to enumerate various filter combinations to find the most accurate solution.

4.3. Multiple Ensemble Combination

For multiple ensembles, the optimal number of base classification algorithms to be combined is hard to establish. This is illustrated by observing the combinations in Table 4b that are tied at rank 2. They have the same measure of accuracy but involve different number of base algorithms. Thus there is no direct relationship between performance and number of algorithms combined.

Strictly combining only the top n classification algorithms from Stage 1 will not always yield the best combination. In Table 4b, the ranked 1 combination combines algorithms at rank 1, 2, and 4 from Table 4a and performs better

than the combination at rank 7 that combines algorithms at rank 1 to 3. Thus, the unique characteristics of each base algorithm subset determines the combination’s accuracy instead of the individual accuracy of each algorithm. These observations justify the need evaluate subsets of the top μ Stage 1 combinations in our methodology.

Symbol	Description
<i>AODE</i>	Averaged One Dependence Estimators
<i>CSMOTE_n</i>	Cluster-Based SMOTE to a ratio of n
<i>OSample_n</i>	Over-sampling to a ratio of n
<i>SMOTE_n</i>	Synthetic Minority Over-sampling TEchnique to a ratio of n
<i>SU_m</i>	Relevance feature selection using Symmetrical Uncertainty with threshold of m
<i>Tomek</i>	Under-sampling both classes using Tomek Links
<i>Vote</i>	Voting

Table 2. Legend for combination Symbols

	F_{0.5}	Combination
1	70.37	<i>CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE</i>
2	66.67	<i>OSAMPLE_{1:2} ▷ SU_{0.04} ▷ AODE</i>
3	63.70	<i>CSMOTE_{1:2} ▷ SU_{0.02} ▷ AODE</i>
4	61.11	<i>OSAMPLE_{1:2} ▷ SU_{0.02} ▷ AODE</i>
5	60.58	<i>OSAMPLE_{1:2} ▷ AODE</i>
6	60.42	<i>CSMOTE_{1:2} ▷ AODE</i>
7	60.37	<i>SU_{0.02} ▷ AODE</i>
8	58.86	<i>SU_{0.04} ▷ AODE</i>

Note: ▷ depicts the sequence of algorithms applied.

Table 3. Aneurysm dataset: comparison of some filter combination algorithms

5. Diagnosis Tool & Prediction Justification

The best combination of algorithms from our experiment (Table 4b) was used in the implementation of an aneurysm rupture prediction tool. Using our proposed methodology, it is very likely that the top combination of algorithms will consist of an ensemble technique. This results in the classification model being unable to provide human understandable reasoning. Hence, we implemented two user functions in the prediction tool that will aid doctors in exploring prediction reasoning.

First, when editing a patient’s record as shown in Figure 5, the prediction tool displays an “on-the-fly” prediction (top-right, circled) based on the current set of values entered in the form. This user function allows doctors to immediately see the effects of varying the values of features that they believe are crucial. Next, a nearest neighbors report (Figure 6) can be generated for each patient record that is based on training instances. This report contains a basic visualization of the class distribution of the ten nearest

	F_{0.5}	Combination
1	70.37	<i>CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE</i>
2	66.67	<i>OSAMPLE_{1:2} ▷ SU_{0.04} ▷ AODE</i>
3	66.49	<i>SMOTE_{1:2} ▷ Tomek ▷ SU_{0.04} ▷ AODE</i>
4	66.32	<i>SMOTE_{1:2} ▷ Tomek ▷ SU_{0.03} ▷ AODE</i>

a) Post stage 1 top 4 combinations

	F_{0.5}	Acc	Combination
1	73.37	92.03	<i>Vote({CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE, OSAMPLE_{1:2} ▷ SU_{0.04} ▷ AODE, SMOTE_{1:2} ▷ Tomek ▷ SU_{0.03} ▷ AODE})</i>
2	71.74	91.55	<i>Vote({CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE, OSAMPLE_{1:2} ▷ SU_{0.04} ▷ AODE, SMOTE_{1:2} ▷ Tomek ▷ SU_{0.04} ▷ AODE, SMOTE_{1:2} ▷ Tomek ▷ SU_{0.03} ▷ AODE, SMOTE_{1:1} ▷ SU_{0.04} ▷ AODE})</i>
2	71.74	91.55	<i>Vote({CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE, SMOTE_{1:2} ▷ Tomek ▷ SU_{0.04} ▷ AODE, OSAMPLE_{1:2} ▷ SU_{0.03} ▷ AODE})</i>
2	71.74	91.55	<i>Vote({CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE, SMOTE_{1:2} ▷ Tomek ▷ SU_{0.03} ▷ AODE, OSAMPLE_{1:2} ▷ SU_{0.03} ▷ AODE})</i>
7	70.88	91.30	<i>Vote({CSMOTE_{1:2} ▷ SU_{0.04} ▷ AODE, OSAMPLE_{1:2} ▷ SU_{0.04} ▷ AODE, SMOTE_{1:2} ▷ Tomek ▷ SU_{0.04} ▷ AODE})</i>

b) Some post Stage 2 combinations showing the need to combine subsets of top algorithm combinations of Stage 1

Table 4. Aneurysm dataset: selected algorithm combinations

neighbors. Also, the features used to evaluate the nearest neighbors can be automatically determined using the current selected features used in prediction. However, although such tools are useful to doctors, we recognize that they do not completely reflect the underlying classification model. More advanced rule extraction techniques may need to be employed in future if robust justification is needed.

6. Conclusion

To the best of our knowledge, this paper introduces the first attempt to use a variety of data mining algorithms to predict ruptures of cerebral aneurysms. We have detailed the challenges faced and presented a generic methodology to find the best combination of algorithms. From the results of experiments on various datasets, we demonstrate the improvement in predicting rare classes, and show that it is not optimal to merely combine the algorithms that perform best on individual tasks. Hence we justify the need to rigorously evaluate combinations to find a near optimal classification algorithm.

Our methodology can be used in other domains that share similar problems of class imbalance and unequal misclassification costs to achieve a mean improvement of 10.2 points in $F_{0.5}$ score. The best combination from our experiment for predicting aneurysm ruptures achieves 92% accu-

racy and is implemented within a prediction tool for use at the hospital together with user functions to aid understanding of prediction reasoning.

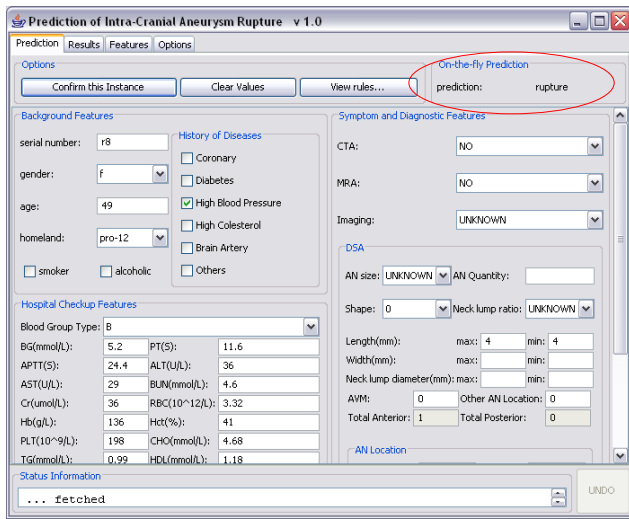


Figure 5. Aneurysm Rupture Prediction Form (on-the-fly prediction circled)

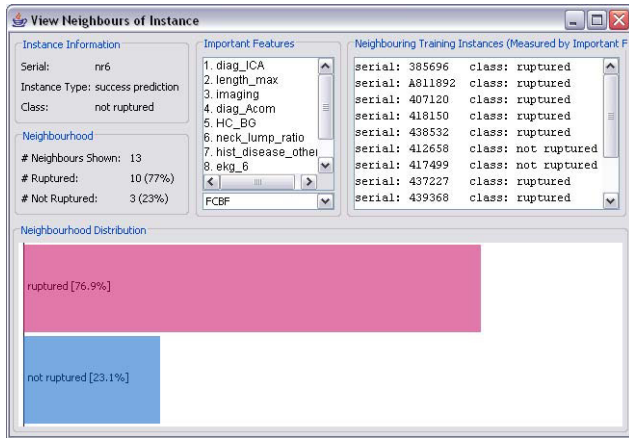


Figure 6. Nearest Neighbors Distribution

References

- [1] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, 2004.
- [2] C. Blake and C. Merz. UCI repository of machine learning database. <http://www.ics.uci.edu/~mlern/MLRepository.html>, 1998.
- [3] L. Breiman. Stacked regressions. *Machine Learning*, 24:41–48, 1996a.
- [4] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

- [5] J. L. Brisman, J. K. Song, and D. W. Newell. Cerebral aneurysms. *The New England Journal of Medicine*, 355(9):928–939, 2006.
- [6] E. Cantú-Paz, S. Newsam, and C. Kamath. Feature selection in scientific applications. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 788–793, New York, NY, USA, 2004. ACM Press.
- [7] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.*, 40(2):139–157, 2000.
- [8] G. A. Donnan and S. M. Davis. Patients with small asymptomatic, unruptured intracranial aneurysms and no history of subarachnoid hemorrhage should be treated conservatively. *Stroke*, 36:407, 2005.
- [9] T. Jo and N. Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explor. Newsl.*, 6(1):40–49, 2004.
- [10] J. Kittler and F. M. Alkoot. Sum versus vote fusion in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):110–115, 2003.
- [11] T. I. S. of Unruptured Intracranial Aneurysm Investigators. Unruptured intracranial aneurysms – risk of rupture and risks of surgical intervention. *The New England Journal of Medicine*, 339(24):1725–1733, 1998.
- [12] D. P. The diameter-cube hypothesis: a new biophysical model of aneurysm rupture. *Surgical Neurology*, 58:166–173(8), September 2002.
- [13] C. Phua, D. Alahakoon, and V. Lee. Minority report in fraud detection: classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1):50–59, 2004.
- [14] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report 98-14, Microsoft Research*, 1998.
- [15] Q. R. J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [16] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.
- [17] G. I. Webb, J. R. Boughton, and Z. Wang. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 58:5–24, 2005.
- [18] G. M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, 6(1):7–19, 2004.
- [19] D. O. Wiebers. Unruptured intracranial aneurysms: natural history, clinical outcome and risks of surgical and edovascular treatment. *The Lancet*, 362:103–110, 2003.
- [20] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.
- [21] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, June 2005.
- [22] C. Yang and S. Létoirneau. Learning to predict train wheel failures. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 516–525, New York, NY, USA, 2005. ACM Press.
- [23] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, 2004.