A combination classification method based on Ripper and Adaboost

Min Wang

College of Finance and Statistics,
Hunan University,
Changsha, Hunan, 410082, China
Email: 2508347005@qq.com

Zuo Chen

College of Computer Science and Electronic Engineering,
Hunan University,
Changsha, Hunan, 410082, China
and
Postdoctoral Collaborative Research and Development Center,
Bank of Changsha,
Changsha, Hunan, 410005, China
Email: chenzuo@hnu.edu.cn

Zhiqiang Zhang

College of Computer Science and Electronic Engineering,
Hunan University,
Changsha, Hunan, 410082, China
Email: 1525065219@qq.com

Sangzhi Zhu* and Shenggang Yang

College of Finance and Statistics,
Hunan University,
Changsha, Hunan, 410082, China
Email: zhusangzhi@hnu.edu.cn
Email: sgyang@hnu.edu.cn
*Corresponding author

Abstract: With the growing demand for data analysis, machine learning technology has been widely used in many applications, such as mass data summarising rules, predicting behaviours and dividing characteristics. The Ripper algorithm presents better pruning and stopping criteria than the traditional decision tree algorithm (C4.5), while its error rate less than or equal to C4.5 by O(nlog2n) time complexity. As a result of that, Ripper can maintain high efficiency even on the massive dataset which contains lots of noise. Adaboost is one of iterative algorithms, which combines a group of weak classifiers together to set up a strong classifier. In order to improve the accuracy of Ripper classification algorithm and reduce the computational complexity, this paper proposes a Ripper-Adaboost combined classification method (Ripper-ADB). The experiment result shows Ripper-ADB could improve the classifier and get higher classification accuracy than decision tree and SVM.

Keywords: Ripper; feature selection; Adaboost; NSL-KDD; C4.5.

Reference to this paper should be made as follows: Wang, M., Chen, Z., Zhang, Z., Zhu, S. and Yang, S. (2021) 'A combination classification method based on Ripper and Adaboost', *Int. J. Embedded Systems*, Vol. 14, No. 3, pp.229–238.

Biographical notes: Min Wang is an Assistant Professor at the College of Finance and Statistics, Hunan University, China. She received her MA from the Hunan University and Nottingham University in 2004 and 2005, respectively. Currently, she is a PhD student in the College of Finance and Statistics, Hunan University, China.

Zuo Chen received his PhD in Hunan University in 2008. He is an Assistant Professor at the College of Computer Science and Electronic Engineering, Hunan University. His research interests include big data, data mining and so on.

Zhiqiang Zhang is studying at the Hunan University, Computer Science and Electronic Engineering. He is a graduate student. He mainly research in big data in the graduate stage at the same time he has also a certain interest in deep learning.

Sangzhi Zhu received his doctoral in the University of Technology Sydney (AU). Now he is currently doing postdoctoral research at the Hunan University in China. His current research interests include financial technology and data mining.

Shenggang Yang is a Professor in Hunan University, China. His research interests are international financial management and credit management, currently focusing on systemic risk, social and economic networks. He received his PhD in Economics from the Wuhan University, China and used to be a Visiting Scholar in National Taiwan University during 1999 and University of Miami during 2006.

1 Introduction

In the big data era, it has become more and more important for applying data mining and machine learning technology in various fields, such as social intelligence assessment and scientific management. In order to improve the quality of data sources, it is a great challenge to select more effective and representative data among a large amount of dataset. Feature selection (Bolón-Canedo et al., 2016; Yu and Liu, 2003) is always used for data dimensionality reduction, by filter (Xue et al., 2012; Maldonado and Weber, 2009) or wrapper (Bermejo et al., 2014) method to select the most representative *d* features from the original dataset with *D* feature attributes, to get an optimised subset of attributes for reducing computational complexity and maintaining high classification accuracy.

Decision tree (Quinlan, 1986) is based on supervised learning with high interpretability which has the advantage over other algorithms. However, tree generation is a process of recursively dividing the whole data space and establishing a local model, which requires high time consuming. Ripper uses a depth-first search strategy to generate rules directly from the dataset instead of to establish a searching tree by making use of the characteristics of the decision tree. After the pruning process, the rules are tailored. Ripper algorithm efficiency is greatly improved classification accuracy.

The Adaboost algorithm (Zhu et al., 2009) is an iterative algorithm and combines these weak classifiers to form a strong classifier. The algorithm is self-adaptive because the samples from the previous base classifier are warped, and the weighted whole sample is used to train the next base classifier again until a predetermined sufficiently small error rate is reached or a pre-specified maximum number of iterations are reached. Adaboost is a highly accurate classifier.

In order to improve the classification accuracy of Ripper algorithm and reduce algorithm complexity, this paper presents a combined classification method based on Ripper and Adaboost (Ripper-ADB). Firstly, by using the wrapper feature selection framework, the feature attributes in the classification rules are weighted. It is iteratively used to achieve the feature selection by filtering out the attributes with smaller weights, and iterating through the method to achieve feature selection. Then we use Ripper as a weak classifier and Adaboost algorithm framework to construct the final classifier.

2 Literature survey

Currently, in the feature selection method, there are basically two types: filter mode and wrapper mode. The filter mode takes some metrics to evaluate the advantages of the feature attributes, commonly such as chi-squared test, information gain, and correlation coefficient scores. However, the wrapper mode directly uses the classifier performance as the evaluation criterion for the feature subset. In general, the efficiency of the filter mode is relatively higher, but the accuracy is always relatively lower than the wrapper mode.

Filter-based feature selection method framework is used to perform feature selection analysis for the KDD CUP 99 dataset with three different algorithms: attribute ranking, attribute scoring, and attribute subsets (Harbola et al., 2014). The attribute ranking algorithm ranks feature attributes according to a ratio of a predefined program, which presents the importance of attributes. The attribute scoring algorithm compresses the score of each attribute between [0, 1], and the attribute whose score is closer to 1 is more important. The efficiency of the classifier can be well improved by these feature selection methods.

SVM classification is based on the conceptual boundaries of decision making. Decision boundaries separate a set of instances with different values into two groups. The SVM classifier establishes a learning mode that assigns new instances to a non-probabilistic binary linear classifier. Taking NSL-KDD Cup 99 as the experimental dataset, a wrapper feature selection algorithm is proposed based on SVM (Han et al., 2017). The experimental results show that the use of fewer feature attributes can achieve a high classification accuracy of 91% in the training set and 99% of the classification accuracy with 36 attributes.

C4.5 decision tree classifier shows complex construction and low classification accuracy (Liu et al., 2011). It is proposed that an improved decision tree construction algorithm based on variable precision rough set (Zhang and Mo, 2004). Compared with the information gain, the approximate classification quality is used as the attribute measurement function of the node selection attribute. It can describe the comprehensive contribution ability of attribute classification more accurately than the information gain. Experimental results show that the decision tree constructed by this algorithm is superior to C4.5 algorithm in classification accuracy and scale.

For the problem that the conditional independence assumption of naive Bayesian classifier does not hold in many cases (Li and Fu, 2016), an algorithm is proposed which is based on principal component analysis combined with naive Bayesian (Wang et al., 2019) classifier. The main idea is to select the features of the original data by principal component analysis, and make the attributes approximate the conditional independence hypothesis, and then construct a naive Bayesian classifier on the new attribute datasets.

Combinatorial classifiers can be divided into two types: bagging and boosting (Tavallaee et al., 2009). The idea of bagging is to use multiple classifiers to train the data sources separately, and use the combined voting method to get the final classification result. Generally, the accuracy of bagging classifier is significantly higher than that of a single classifier. Boosting focuses on the iterative training methods and update each classifier voting weights which is a function of accuracy. Compared with the bagging method, the boosting method is more flexible and accurate in the training process.

There is an iterative combination classifier based on the SVM classification algorithm combined with the Adaboost algorithm framework (Li et al., 2008). Adaboost algorithm has certain limitations on the establishment of classifiers for rare samples. This method increases the weight of rare samples in sampling, making it easier to extract in the iterative process, so as to make the classifier hard to ignore the rare samples.

An improved Adaboost algorithm based on decision tree is proposed and applied to malicious android applications (Chen et al., 2018). In the traditional cascade decision, each classifier only considers the result of the current classifier's decision on the sample, but does not consider the result of the previous classifier. In this research, for each level of weak classifier, an additional discriminate function is added. The experimental results show that the Adaboost-DT algorithm is different from the traditional Adaboost algorithm based on SVM. By using C4.5 decision tree algorithm as a weak classifier, the efficiency of Adaboost algorithm is significantly improved.

In this paper, we propose a combination classification method based on Ripper and Adaboost. We choose the wrapper feature selection algorithm framework to combine with the characteristics of the classification algorithm. According to the classification rules generated by the Ripper algorithm, the weights of the attributes in the rules are calculated. After filtering out attributes with small weights iteratively, it could achieve the purpose of feature selection. We set up an Adaboost-based structure to use the combined iterative method to make multi-classifiers more flexible and efficient, so as to generate a final high-performance strong classifier.

3 Ripper-ADB combination classification algorithm

3.1 Data preprocessing

3.1.1 Algorithm of Ripper

Ripper (Shang et al., 2013) is a rule-based classification algorithm. Each Ripper rule consists of a set of rule antecedents that includes better pruning and stopping criteria and handling of the rule set. This is an incremental reduction error pruning algorithm that splits instances of a training set into two datasets, a growth set and a pruning set. The growth set is used to generate rules and add conditions until the rules are perfect. Pruning set is used to build rules, delete the rules, until getting better rules. Then evaluate the value of the rules, remove the last conditions to see whether the value changes or not. If there is no change, continue to remove the conditions until the best result is obtained.

Ripper algorithm implementation process is as follows:

1. Divide the data items that do not belong to the rules (in the training data) into two subsets randomly – the growth set and the shrink set.
2. Rules of the expansion process. First set the rules of the conditions empty, then join the formula conditions:
   $$A_d = v,\ A_n \le \theta \text{ or } A_n \ge \theta \quad (1)$$
3. A_d is a character-type attribute; v is a valid value of A_d; A_n is a real variable, θ is the effective value of A_n appearing in the training set. So repeatedly add conditions to the rules, so that information gain (D, At) (Keogh and Mueen, 2011) reach a greater value.
4. Rule reduction process. The last condition is removed from the rule conditions in order to maximise the function value.

The Ripper algorithm does not need to establish a complete decision tree in advance, so that its efficiency is higher than C4.5. The Ripper algorithm has obvious advantages in terms of noise data processing ability, learning efficiency and knowledge comprehension.

3.1.2 Feature selection based on Ripper

The size of the input dataset for a classification task can be described by two parameters: N and P. KDD analysis data is always very large in N and P, resulting in 'dimensionality disaster (Miller et al., 2013)' and 'combination explosion'.

Due to the reduction of the number of features, it is also possible to eliminate some repetitive instances and reduce numbers of P. This effectively resolves the problems of dimension disasters and combined explosions. Due to the decrease of N and P, the algorithm learning time can be reduced and the classification efficiency can be improved at the same time.

Firstly, the training set of NSL-KDD 99, which has 41 characteristic attributes initially, is trained by using Ripper algorithm, the weight of each attribute in the classification rule is calculated, the feature attribute with smaller weight is filtered, then use the new feature dataset training classifier, iteration for feature attribute filtering. As shown in Figure 1.

Figure 1 Feature selection algorithm based on Ripper

```
Algorithm 1 Feature Selection
Input: D,K    //Original data set:D.The number of attributes to keep:K.
Output: S     //The data set after the filter:S.
Algorithm:
  Si=D
  while Ture:
  //Construct Si attribute Ripper classifier, get the result of classification rule Ci.
  for eachFeature in Ci:
    //Statistics each attribute weights, generate dictionary Di.
    Di[eachFeature]=Ci.count(eachFeature)
  end for
  for eachFeatureWeight in Di:
  //If the weight is less than the given threshold, delete this property.
    if eachFeatureWeight < threshold_value:
      Si.remove(Feature);
    end if
  end for
  if count(Si) == K:
    //If the number of attributes is equal to k,End the cycle.
    break;
  end if
end while
```

The algorithm is as follows:

1 Enter the original dataset D, the required number of features K.

2 Construct Si attribute Ripper classifier (initial Si = D), resulting in classification rules Ci.

3 The weight of each attribute in Ci is calculated and stored in the dictionary Di.

4 Traverse the weight of each attribute, if less than the threshold, then delete the attribute in Si.

5 If the number of feature attributes is equal to K. The result Si is outputted. Otherwise, jump to step 2.

According to the above steps, we can filter the feature attributes one by one to get the most valuable and most representative feature attributes, so as to reduce the complexity of the operation.

3.2 Combination classification algorithm

3.2.1 Adaboost algorithm principle introduction

The adaptation of Adaboost is that the samples of the previous weak classifier are warped, and the weighted samples are reused to train the next basic classifier. At the same time, add a new weak classifier in each round until a predetermined small enough error rate is reached.

Specifically, the weight distribution of the training data is initialised first. During training, if a sample has been accurately classified, its weight is reduced when constructing the next training set. Conversely, if a sample is not accurately classified, its weight is increased. The updated sample of weights is then used for the next classified. The weak classifiers obtained by each training are combined into a strong classifier. After the training process of each weak classifier is completed, the weight of the weak classifier with smaller classification error rate is increased, so that it reduces the weight of the weak classifier with large classification error rate. The specific process is as follows:

1 First, we initialise the weight distribution of the data. Each training sample is given the same weight: $1/N$.

$$D_1 = \left(W_{11}, W_{12}, \cdots, W_{1i}, \cdots, W_{1N}\right), \quad W_{1i} = \frac{1}{N}, i = 1, 2, \cdots, N \tag{2}$$

D_1 denotes a weighted training set sample set; w represents the weight of each training sample; N indicates the total number of sample sets and i represents the count of each sample.

2 Perform multiple rounds of iteration, using $m = 1,2, ..., M$ to represent iteration rounds.

a Using a training dataset with weight distribution D_m, get the basic classifier:

$$G_m(x): \chi \rightarrow \{-1, +1\} \tag{3}$$

$G_m(x)$ denotes the m^{th} classifier and χ indicates the classifier distribution range.

b Calculate the classification error rate of $G_m(x)$ on the training dataset:

$$e_m = P\left(G_m\left(x_i\right) \neq y_i\right) = \sum_{i=1}^{N} W_{mi} I\left(G_m\left(x_i\right) \neq y_i\right) \tag{4}$$

P is the error probability value calculation expression; $G_m(x_i)$ represents the prediction result of the m^{th} classifier for the i^{th} sample; y_i indicates the classification result of the i^{th} sample; I indicates judgment condition and w represents the weight of the wrong sample.

c Calculate the coefficient of $G_m(x)$, and α_m represents the importance of $G_m(x)$ in the final classifier (purpose: to get the weight of the basic classifier in the final classifier weight).

d The training dataset is updated for weight distribution (purpose: to get the new weight distribution for the sample) for the next iteration.

$$D_{m+1}\left(W_{m+1,1}, W_{m+1,2}, \cdots, W_{m+1,i}, \cdots, W_{m+1,N}\right) \tag{5}$$

$$W_{m+1,i} = \frac{W_{mi}}{Z_m}\exp\left(-\alpha_m y_i G_m(x_i)\right), i = 1, 2, \cdots, N \tag{6}$$

Z_m is the normalisation factor; make D_{m+1} a probability distribution. So that the weight of the sample misclassified by the basic classifier $G_m(x)$ increases, while the weight of the sample correctly classified decreases. In this way, the AdaBoost method can 'focus on' the more difficult samples.

$$Z_m = \sum_{i=1}^{N} w_{mi}\exp\left(-\alpha_m y_i G_m(x_i)\right) \tag{7}$$

3 Combine each weak classifier:

$$f(x) = \sum_{m=1}^{M}\alpha_m G_m(x) \tag{8}$$

This gives the final classifier as follow:

$$G(x) = sign\left(f(x)\right) = sign\left[\sum_{m=1}^{M}\alpha_m G_m(x)\right] \tag{9}$$

sign is a symbol function, that is, when the function value is greater than 0, the result is 1, and when it is equal to 0, the result is 0. When the function value is less than 0, the result is –1.

According to Adaboost algorithm and Ripper classifier, this paper constructs Ripper-ADB classification method.

3.2.2 Ripper-ADB classification enhancement method

Based on the above Adaboost algorithm framework, this paper constructs three classifiers with different iterations. Adaboost's main idea is to increase the weight of its samples for misclassified data, while reduce the sample weight by 50% to correctly classifying the data. However, the traditional training method has two disadvantages:

1 If all the samples are used for iterative training, the number of samples will increase exponentially after each iteration, more difficulty to keep training.

2 If using random sampling to form the corresponding weight ratio, some samples will be missed or ignored, resulting in incomplete training.

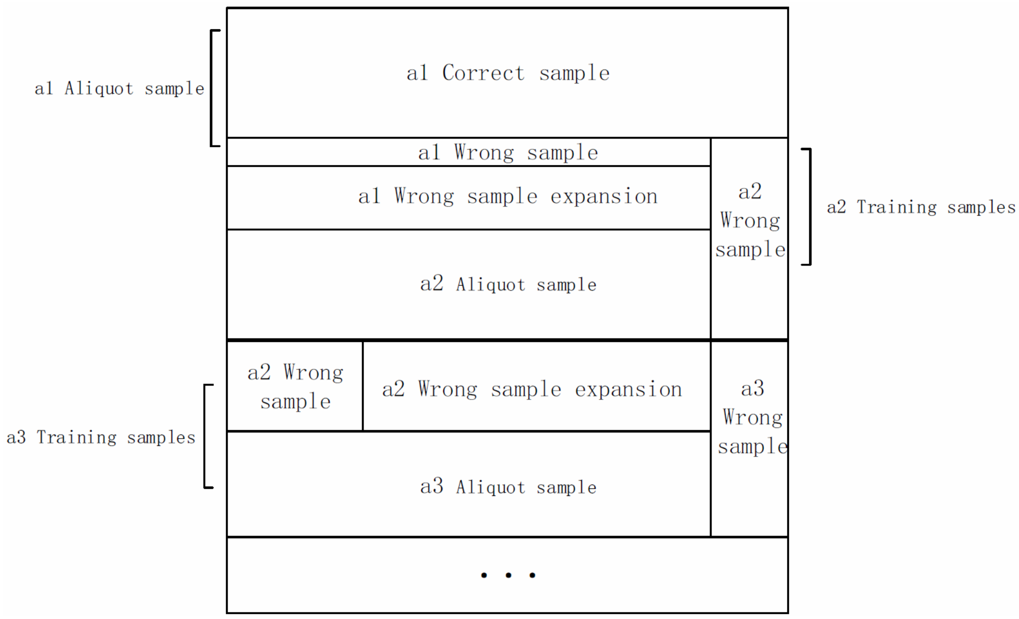
Consider the above two points, using a training method shown in Figure 2 for cyclic superposition samples.

Firstly, the training set is divided into equal parts according to the number of iterations, and then the first sample is trained using the Ripper algorithm to obtain the classifier a1. Then a1 classifier mispredicted data samples were expanded to equal the number of the next sample (50% each) to get a new data sample. The newly generated data samples serve as a2 classifier training data. Loop this process iteratively until all the samples have been trained. Finally, the weights of different classifiers are integrated to form strong classifiers.

In the above loop iteration method, each sample can be trained, and superposition training is performed on the data of the iterative misclassification.

Finally, combined with the preprocessing feature selection method, the construction of Ripper-ADB classifier is carried out with different number of feature attributes.

Figure 2 Cyclic training chart



4 Experimental results and analysis

4.1 NSL-KDD99 dataset introduction

KDD is an abbreviation for data mining and knowledge discovery. KDD Cup 99 (Tavallaee et al., 2009) dataset is a dataset evaluated by Defense Advanced Research Projects Agency (DARPA) IDS and used for KDD Cup 1999 game data. The complete dataset has nearly 5 million connection records, each with 41 signature attributes, and the exception types are subdivided into 39 categories in four major categories.

Based on the above-mentioned large datasets, Mahbod Tavallaee found a number of important issues affecting the performance of the system when it came to collect data statistics. In order to solve these problems, they proposed a new dataset NSL-KDD (Ding et al., 2017), which has the following advantages for the original KDD:

1 does not include duplicate records on the training set

2 does not include duplicate records on the test set

3 the type of choice is representative

4 it is reasonable to choose the number of training and test sets.

NSL-KDD 99 is used as the experimental dataset, based on the above characteristics of NSL-KDD 99 and the test set of NSL-KDD 99, which are extremely rigorous.

4.2 Feature selection preprocessing results

Feature attribute selection algorithm used the original dataset of 41 feature attributes first. For the first time, filtered out the characteristic attributes that have not appeared (weight is 0), we got 32 selected characteristic attributes. Then using the training set with the new feature attributes to classify training in the classifier, generated a new classification rule. Repeat the above steps to achieve the purpose of feature selection. The experimental results are shown in Table 1.

In the meantime, ten-fold crossover test is performed on the training set (Luo et al., 2017) [Ten-fold cross-validation divides the training data into ten sub-datasets (N/10) and then uses nine datasets to train and the remaining one subset for testing. The process is repeated ten times to get the average classification accuracy.]. We compare the previous article with SVM-based feature selection algorithms. Figure 3 and Figure 4 shows the classification accuracy rate comparison between the Ripper and SVM on training sets, the ten-fold cross-validation of the training set, and the feature number of the test set separately. The SVM-based feature selection algorithms as shown in Figure 6. Table 5 shows the comparative experimental results.

In the meantime, ten-fold crossover test is performed on the training set (Luo et al., 2017) [Ten-fold cross-validation divides the training data into ten sub-datasets (N/10) and then uses nine datasets to train and the remaining one subset for testing. The process is repeated ten times to get the average classification accuracy.). We compare the previous article with SVM-based feature selection algorithms. Figure 3 and Figure 4 shows the classification accuracy rate comparison between the Ripper and SVM on training sets, the ten-fold cross-validation of the training set, and the feature number of the test set separately. The SVM-based feature selection algorithms as shown in Figure 6. Figure 4 shows the comparative experimental results.

Table 1 Feature attribute weight filter results

The number of features	*Filter features*	*Weight*
32 weight $\leqslant 2$	land	1
	wrong_fragment	2
	num_failed_logins	1
	logged_in	2
	num_compromised	0
	root_shell	2
	num_shells	1
	srv_serror_rate	2
	dst_host_srv_serror_rate	2
	protocol_type	5
	hot	5
23 weight $\leqslant 5$	serror_rate	3
	same_srv_rate	5
	dst_host_same_srv_rate	5
	dst_host_serror_rate	1
17 weight $\leqslant 6$	num_file_creations	4
	srv_diff_host_rate	6
	dst_host_srv_diff_host_rate	5
	Duration	6
14 weight $\leqslant 11$	dst_bytes	11
	rerror_rate	11
	dst_host_rerror_rate	9
10 weight $\leqslant 27$	flag	22
	dst_host_diff_srv_rate	27
	dst_host_same_src_port_rate	18
7 weight $\leqslant 12$	diff_srv_rate	12

From Figures 3 and 4, it can be seen that Ripper's algorithm has higher classification accuracy, than the SVM algorithm when a large number of initial feature attributes are available. With the use of the feature selection algorithm proposed in this paper, it can be clearly seen that when the number of features is between 36 and 17, SVM-based feature selection algorithm has a slower downward trend, while Ripper-based feature selection algorithm can still maintain higher classification accuracy. When feature attributes are filtered out to 17 attributes, both feature selection algorithms have a significant downward trend. However, due to the algorithm proposed in this paper, combined with the characteristics of the classifier itself, it is more stable. The accuracy rate is obviously higher than the SVM feature selection algorithm.

Figure 5 shows the experimental comparison of our algorithm and the SVM feature selection algorithm on the test set. Due to the particularity of the NSL-KDD 99 test set, this does not contain training set data, so there will be appropriate fluctuations. However, it can be seen that the feature selection algorithm in this paper is higher than the SVM algorithm in classification accuracy.

Figure 3 Classification accuracy employing SVM and Ripper with selected features on training data (see online version for colours)

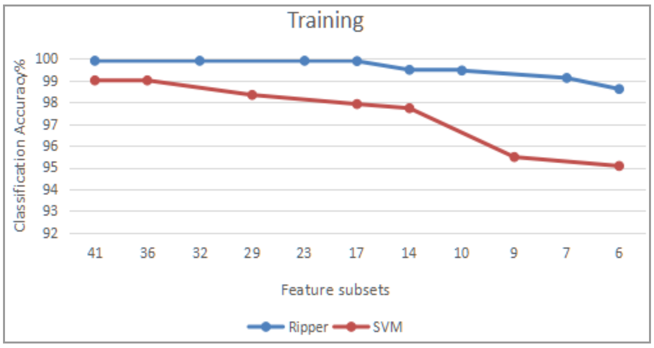


Figure 4 Classification accuracy employing SVM and Ripper with selected features using on ten-fold cross-validation (see online version for colours)

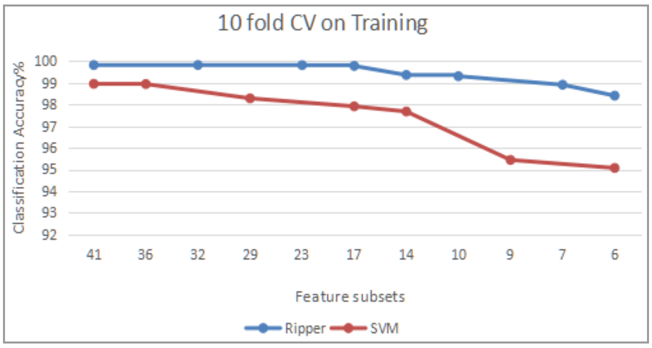


Figure 5 Classification accuracy employing SVM and Ripper with selected features on test data (see online version for colours)

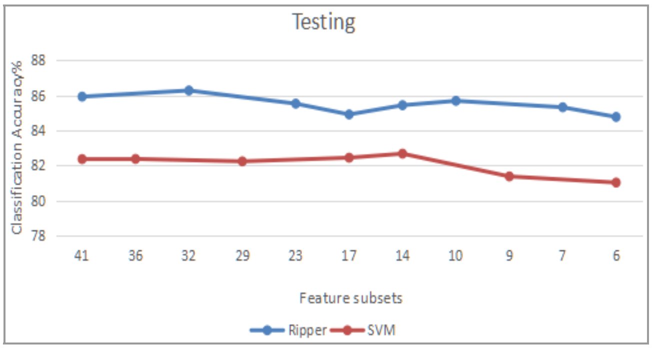


Figure 6 Classification accuracy employing Ripper-ADB and Ripper with selected features on training data (see online version for colours)

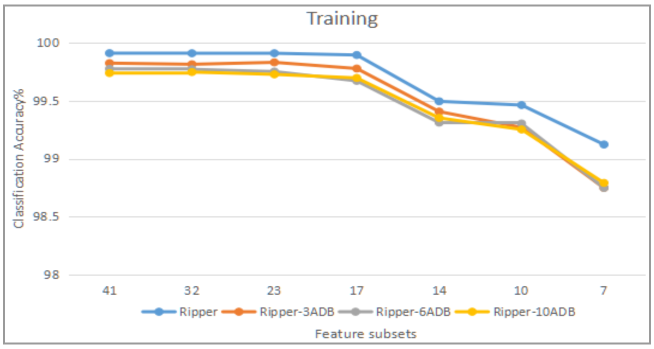


Figure 7 Classification accuracy employing Ripper-ADB and Ripper with selected features on test data (see online version for colours)

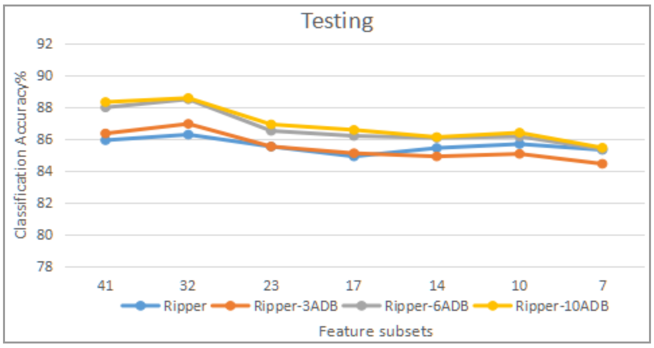


Figure 8 Classification accuracy employing Ripper-ADB, J48 and SVM with selected features on training data (see online version for colours)

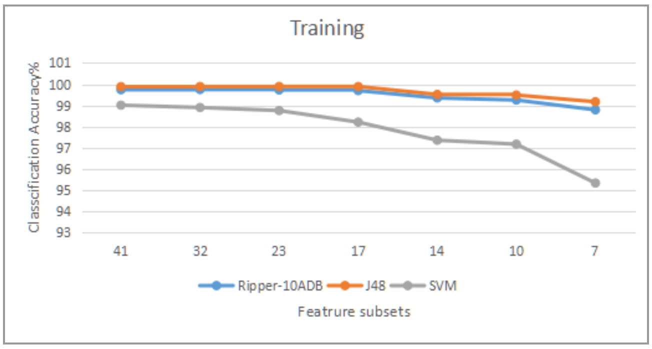


Table 2 shows a comparison of classification accuracy over the test set with valid feature selection. Reflecting the reduction of feature attributes, combining with the characteristics of the classifier, the corresponding feature attribute selection can maintain the high accuracy better.

Table 2 Classification accuracy employing SVM and Ripper with selected features

Features	*Ripper*			*SVM*		
	Training	*Ten-fold CV on training*	*Testing*	*Training*	*Ten-fold CV on training*	*Testing*
41	99.91	99.83	85.94	99.01	98.96	82.37
36	\	\	\	99.01	98.95	82.38
32	99.91	99.83	86.29	\	\	\
29	\	\	\	98.34	98.29	82.24
23	99.91	99.82	85.54	\	\	\
17	99.90	99.79	84.92	97.92	97.92	82.45
14	99.50	99.37	85.45	97.73	97.68	82.68
10	99.46	99.31	85.70	\	\	\
9	\	\	\	95.48	95.44	81.38
7	99.12	98.92	85.33	\	\	\
6	98.61	98.41	84.78	95.07	95.01	81.03

4.3 Ripper-ADB algorithm experiment results

According to the characteristics of the selected combination of Ripper-ADB algorithm, the three classifier, six classifier and ten classifier experimental data division details of the experimental comparison with the original algorithm is shown below.

1 Ripper-3ADB
  a The original data is roughly divided into three parts, each about 40,000 data volumes.
  b Training the classifier a1 for the first part of data.
  c For classifier a1 sub-wrong data, the misclassified data is 50% resampled and filled into the second part of data to form about 80,000 training data. And training the classifier a2 for the 80,000 data.

2 Ripper-6ADB
  a The original data is roughly divided into six parts, each about 20,000 data volumes.
  b Training the classifier a1 for the first part of data.
  c For classifier a1 sub-wrong data, the misclassified data is 50% resampled and filled into the second part of data to form about 40,000 training data. And training the classifier a2 for the 40,000 data.

3 Ripper-10ADB
  a The original data is roughly divided into ten parts, each about 12,000 data volumes.
  b Training the classifier a1 for the first part of data.
  c For classifier a1 sub-wrong data, the misclassified data is 50% resampled and filled into the second part of data to form about 24,000 training data. And training the classifier a2 for the 24,000 data.

It can be seen from Figure 6 that the classification accuracy of four classifiers begins to drop significantly after the 17 attributes due to the reduction of the characteristic attributes. Due to Ripper's one-time training of full datasets, it leads to better fit to all the data and the highest classification accuracy. The Ripper-ADB, Ripper-6ADB and Ripper-10ADB, the three curves are similar. When the classifier iteration, the number of samples per sample decreases, each classifier has a specific choice. Classifiers can play a good balance of not falling in over-fitting training data.

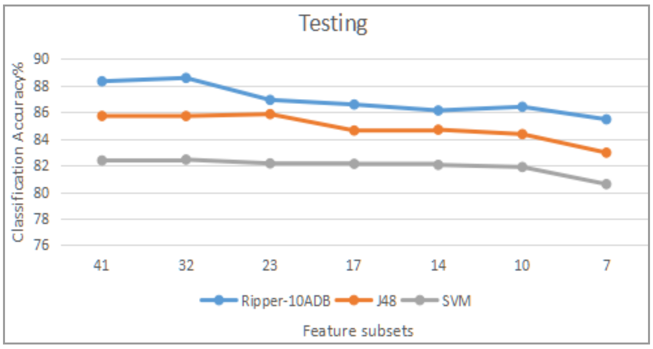
As can be seen from Figure 7, in the Ripper-3ADB classifier, the classification accuracy that is basically consistent with Ripper is reduced until the feature attribute is reduced to 17 attributes. After that, the three classifier has a limited effect on balance due to the reduction of attributes. When the difference of classifier weights is small, the two classifiers have the absolute right to judge, resulting in lower classification accuracy. The Ripper-6ADB, Ripper-10ADB, at the beginning of the 41 attributes and 32 attributes can reach about 88% of the classification accuracy, when the attribute dropped to the accuracy is also higher than the original classifier. This is due to the fact that more classifiers produce good balance effects and play a significant role in the cumulative training of the misclassified training datasets.

In summary, we use the higher efficiency Ripper-10ADB under the same training set, test set compared with the traditional machine learning algorithm SVM, decision tree (J48 is the implementation of decision tree algorithm C4.5 in WEKA software). The result is shown below.

From Figure 8, it can be seen that in the training set, the Ripper-ADB algorithm proposed in this paper and the decision tree (C4.5) almost have consistent classification accuracy in the process of gradually decreasing attributes. SVMs have low classification accuracy because they are difficult for large-scale data processing and cannot efficiently process data containing many noises.

According to the experimental results of the test set shown in Figure 9, due to the Ripper-ADB multi-classifier effect and the erroneous data repetitive training process, it does not appear to overfit the training set with high classification accuracy. For C4.5 and SVM algorithms, the highest classification accuracy of 88.5814% was achieved in 32 attributes.

Figure 9 Classification accuracy employing Ripper-ADB, J48 and SVM with selected features on test data (see online version for colours)



5 Conclusions

Datasets of machine learning applications are always over-dimensioned and have high computational complexity. In order to eliminate redundant attributes as much as possible and improve the accuracy of the classifier, the feature of NSL-KDD99 dataset was screened by using the iterative Wrapper feature selection algorithm. Then we proposed a Ripper-ADB classification algorithm based on Ripper algorithm and Adaboost algorithm framework. The proposed Ripper-ADB algorithm can accumulate enough training for the misclassification data during the iteration of ten classifiers, and get higher classification accuracy results. In the experimental results, the SVM-based feature selection method is mentioned in the article, and the Wrapper feature selection method of loop iteration makes the classification accuracy higher. Using the above feature selection algorithm, combined with the Ripper-ADB algorithm, it can effectively avoid over-fitting on the training set, and achieves a classification accuracy of up to 88.5814% on the test set. Compared with some traditional classification algorithms, such as decision tree, SVM, and the original algorithm Ripper, Ripper-ADB could make the accuracy of the classifier improved.

Considering the Adaboost algorithm framework, there is no reasonable automatic judgment function for outliers when raising the weight of misclassified data during each iteration. So in future work, we will start with the wrong samples in each iteration of Adaboost and further study the method of automatically judging outliers.

Acknowledgements

This work was sponsored by National Natural Science Foundation of China (71790593), Postdoctoral Science Foundation of China (2018M642967) and Major Issues of Social Sciences Fund in Hunan of China (18ZWA15).

References

Bermejo, P., Gámez, J.A. and Puerta, J.M. (2014) 'Speeding up incremental wrapper feature subset selection with naive Bayes classifier', *Knowledge-Based Systems*, Vol. 55, pp.140–147.

Bolón-Canedo, V., Sánchez-Maroño, N. and Alonso-Betanzos, A. (2016) 'Feature selection for high-dimensional data', *Progress in Artificial Intelligence*, Vol. 5, No. 2, pp.65–75.

Chen, X., Mu, R. and Yan, Y. (2018) 'Automated identification of callbacks in Android framework using machine learning techniques', *International Journal of Embedded Systems (IJES)*, Vol. 10, No. 4, pp.301–322.

Ding, F., Ge, Q., Jiang, D., Fu, J. and Hao, M. (2017) 'Understanding the dynamics of terrorism events with multiple-discipline datasets and machine learning approach', *PLoS One*, Vol. 12, No. 6, p.e0179057.

Han, Z., Yang, K., Xu, F. and Duan, D. (2017) 'Probabilistic graphical model for detecting spammers in microblog websites', *International Journal of Embedded Systems (IJES)*, Vol. 8, No. 1, pp.12–23.

Harbola, A., Harbola, J. and Vaisla, K.S. (2014) 'Improved intrusion detection in DDoS applying feature selection using rank & score of attributes in KDD-99 data set', *International Conference on Computational Intelligence and Communication Networks*, pp.840–845.

Keogh, E. and Mueen, A. (2011) *Curse of Dimensionality*, Springer, USA.

Li, C. and Fu, Z. (2016) 'Improvement of simple bias classifier', *Statistics and Decision*, Vol. 5, No. 21, pp.9–11.

Li, Y., Liu, X. and Chen, P. (2008) 'Combined classification algorithm based on improved AdaBoost algorithm and SVM', *Computer Engineering and Application*, Vol. 44, No. 32, pp.140–142.

Liu, X., Wang, L. and Chen, F-x. (2011) 'Improved C4.5 decision tree algorithm based on variable precision rough set', *Computer Application Research*, Vol. 28, No. 10, pp.3649–3651.

Luo, H., Wan, C. and Guo, M. (2017) 'Fusing multiple features and spatial information for image classification via codebook ensemble', *International Journal of Embedded Systems (IJES)*, Vol. 9, No. 3, pp.229–240.

Maldonado, S. and Weber, R. (2009) 'A wrapper method for feature selection using support vector machines', *Information Sciences*, Vol. 179, No. 13, pp.2208–2217.

Miller, F.P., Vandome, A.F. and Mcbrewster, J. (2013) 'Combinatorial explosion', *Encyclopedia of Operations Research & Management Science*, Vol. 20, No. 3, pp.192–192.

Quinlan, J.R. (1986) 'Induction on decision tree', *Machine Learning*, Vol. 1, No. 1, pp.81–106.

Shang, C., Li, M., Feng, S. et al. (2013) 'Feature selection via maximizing global information gain for text classification', *Knowledge-Based Systems*, Vol. 54, No. 4, pp.298–309.

Tavallaee, M., Bagheri, E., Lu, W. et al. (2009) 'A detailed analysis of the KDD CUP 99 data set', *IEEE International Conference on Computational Intelligence for Security & Defense Applications*, IEEE, pp.1–6.

Wang, C., Zhao, Y. and Mo, X. (2019) 'A detection model of malicious android applications based on naive Bayes', *International Journal of Embedded Systems (IJES)*, Vol. 11, No. 4, pp.508–515.

Xue, B., Cervante, L., Shang, L. et al. (2012) 'A multi-objective particle swarm optimisation for filter-based feature selection in classification problems', *Connection Science*, Vol. 24, Nos. 2–3, pp.91–116.

Yu, L. and Liu, H. (2003) 'Feature selection for high-dimensional data: a fast correlation-based filter solution', in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp.856–863.

Zhang, X. and Mo, Z. (2004) 'Variable precision rough sets', *Pattern Recognition and Artificial Intelligence*, Vol. 17, No. 2, pp.151–155.

Zhu, J., Zou, H., Rosset, S. et al. (2009) 'Multi-class AdaBoost', *Statistics & Its Interface*, Vol. 2, No. 3, pp.349–360.