Improving Performance Benchmark of Decision Tree Classifications for E-mail Spam Filtering

Nguyen Van Truong, Doan Thi Minh Thai, Le Bich Lien, Nguyen Thi Thu

***Abstract*— E-mail spam is one of the biggest challenges for Internet users today. It causes a lot of troubles to people and does indirect damages to the economy. This paper presents three approaches to improve e-mail spam detection performance of tree classifications in term of quality: parameter fining-tune, feature selection, and combination of classifiers. The experimental data is a popular dataset called Spambase for the classification of spam. The spam e-mails were classified utilizing 10 fold cross validation by using Weka machine learning software involving 12 different decision trees. The experimental results show that our approaches are a competitive solution to the problem.**

***Index Terms*— Spam filtering, Spam E-mail Detection, Spam Detection Technologies, Machine Learning.**

I. INTRODUCTION

E-mail is one of the most popular means of communication nowadays. There are billions of e-mails sent every day in the world, half of which are spams. Spams are unexpected e-mails for most users that are sent in bulk with main purpose of advertising, stealing information, spreading viruses. For example, Trojan.Win32.Yakes.fize is the most malicious attachment Trojan that downloads a malicious file on the victim computer, runs it, steals the user's personal information and forwards it to the fraudsters [1].

There are a lot of spam filtering methods such as Blacklisting, Whitelisting, Heuristic filtering, Challenge/Response Filter, Throttling, Address obfuscation, and Collaborative filtering. Complicated techniques are used to improve accuracy affects the speed of the whole system as well as the psychology of users [2].

A lot of machine learning-based research works have been presented concerning the classification problems such as Naïve Bayes, Support Vector Machine, K-Nearest Neighbors, and decision trees. These involve using classification algorithms, software tools, datasets, and classification accuracies. Examples of such published efforts are briefly mentioned as follows.

Sharma et al. conducted an experiment using the WEKA environment by handling four classification algorithms namely ID3, J48, Simple Classification And Regression Tree (CART), and Alternating Decision Tree on the spam e-mail dataset [2]. Such classification algorithms are used to categorize the e-mails as spam or non-spam. The algorithms are analyzed and compared in terms of classification accuracy. From the results it was found that the highest accuracy performance is reached by J48 classifier for the spam e-mail datasets Spambase.

Akçetin et al. determined the most convenient decision tree method in terms of accuracy and classification built time by comparing the performance of decision tree algorithms to identify the spam e-mails [3]. Their work shows the process of Weka analysis on tree classifications. Experimental results showed that Random Forest algorithm was the best classifier with the accuracy rate of 94.68% on Spambase dataset [4].

Elhamayed compared the performance of three classification methods Classification and Regression Tree, K-Nearest Neighbour, and Principal Component Analysis [5]. The author calculated the significant parameters which have a direct effect on the performance of the classification methods.

Truong el al. proposed a method of combining positive selection and negative selection in Artificial immune systems for e-mail spam detection. The method is un-supervised machine learning approach with different optimal values of matching lengths [6]. They obtained a remarkable classification performance on both an e-mail corpus and a SMS one.

Iqbal el al. examined learning methods to analyze the strength and weakness of current technologies for spam detection [7]. They noticed that, high performance with minimum features was achieved with both J48 and Naïve Bayes algorithms on imbalanced dataset.

The organization of this paper is as follows: Section 2 presents the proposed methods. Section 3 briefly describes our experiments and discusses the results. Section 4 concludes the whole work.

II. PROPOSED METHODS

This section presents three approaches to obtain an increase in accuracy, that are parameters fine-tuning or parameters optimization, feature selection or attribute selection, and classifiers combination.

We used Weka, a suite of machine learning free software developed at the University of Waikato, New Zealand, for implement our methods and experiments.

A. Parameters Optimization

To get the most out of a machine learning algorithm we can tune the parameters of the method to our problem. We cannot know how to best do this beforehand, therefore we must try out lots of different parameters. The Weka Experiment Environment allows us to design controlled experiments to compare the results of different algorithm parameters and whether the differences are statistically significant.

Since finding the optimal parameters for a classifier can be a rather tedious process, Weka offers some ways of

Nguyen Van Truong, Faculty of Mathematics, Thai Nguyen University of Education, Thai Nguyen City, Vietnam, Mobile No. +(84)915016063,
Doan Thi Minh Thai, Faculty of Mathematics, Thai Nguyen University of Education, Thai Nguyen City, Vietnam, Mobile No. +(84)968158818,
Le Bich Lien, Faculty of Mathematics, Thai Nguyen University of Education, Thai Nguyen City, Vietnam, Mobile No. +(84)983444586,
Nguyen Thi Thu, Faculty of Mathematics, Thai Nguyen University of Education, Thai Nguyen City, Vietnam, Mobile No. +(84)978875752

automating this process a bit. The following meta-classifiers allow you to optimize some parameters of your base classifier: CVParameterSelection, GridSearch, MultiSearch, Auto-WEKA. After finding the best possible setup, the meta-classifiers then train an instance of the base classifier with these parameters and use it for subsequent predictions. Brief introduction for parameter fine-tuning can be found in [8].

B. Feature Selection

Raw machine learning data contains a mixture of attributes, some of which are relevant to making predictions. Therefore we want to know which features to use and which to remove. The process of selecting features in data to model considered problem is called feature selection. It is also called variable selection or attribute selection.

Many feature selection techniques are supported in Weka. The process is divided into two parts: Attribute Evaluator and Search Method. Each section has multiple techniques from which to choose.

C. Classifiers Combination

The method has various names: ensemble method, committee, classifier fusion, combination, aggregation, etc. Combined algorithms are a powerful class of machine learning algorithm that combine the predictions from multiple models. A benefit of using Weka for applied machine learning is that makes available so many different ensemble machine learning algorithms such as Voting and Stacking. They are different from each other in terms of how it works and key algorithm parameters.

Voting is perhaps the simplest ensemble algorithm, and is often very effective. It can be used for classification or regression problems. Voting works by creating two or more sub-models. Each sub-model makes predictions which are combined in some way, such as by taking the mean or the mode of the predictions, allowing each sub-model to vote on what the outcome should be.

Stacking for short is a simple extension to Voting ensembles that can be used for classification and regression problems. In addition to selecting multiple sub-models, stacking allows you to specify another model to learn how to best combine the predictions from the sub-models. Because a meta model is used to best combine the predictions of sub-models, this technique is sometimes called blending, as in blending predictions together.

Brief introduction about other top ensemble machine learning algorithms as well as about key configuration parameters for ensemble algorithms in Weka can be found in [9], [10].

III. Experiments and Discussion of Results

In this section we compare the proposed methods' results of the 12 decision tree algorithms namely Random Forest, Logistic Model Tree (LMT), Functional Tree (FT), J48 Graft, Naive Bayes Tree (NBTree), J48, Reduced Error Pruning Tree (REPTree), Best-First Decision Tree (BFTree), Simple Cart, Alternating Decision Tree (ADTree), LogitBoost Alternating Decision (LADTree), and Random Tree.

As in previous work [3], we use a common performance measurement Overall Accuracy (ACC) to compare performance of some methods. We use a computer with CPU Pentium P6200 2.13GHz, RAM 2GB for all experiments.

A. Dataset

The experimental data is a popular dataset called Spambase for the classification of spam. The Spambase data set was created by Mark Hopkins et al. at Hewlett-Packard Labs. In this dataset the number of instances is 4601 out of which 1813 Spam which is equal 39.4% and the numbers of attributes are 58 out of which 57 are continuous and 1 has nominal class label [4].

B. Experiments For Parameters Optimization

The parameters optimization is processed by examined possible ranges of features for each algorithm. The ranges, iterative steps, and found optimal values are listed in the 3rd column and in the last column of Table 1, respectively. The corresponding accuracy improvements is in Table 2.

Table1. Examined ranges and found optimal values of parameters

Algorithm	Feature		
	Name	Range	Optimal value
Random Forest	maxDepth	(0)	0
	numFeatures	(1;20, 20)	2
	numTrees	(10;100, 18)	100
	seed	(0;100, 11)	1
LMT	numBoostingIterations	(-5;2, 7)	0
	minNumInstances	(10;30, 5)	15
	weightTrimBeta	(0.0;0.9, 10)	0.0
FT	numBoostingIterations	(1;20, 20)	8
	minNumInstaces	(1;50, 10)	15
	weightTrimBeta	(0.0;0.9, 10)	0
J48graft	confidenceFactor	(0.1;0.3, 30)	0.25
	minNumObj	(1;10, 10)	1
NBTree	Debug	(True; False)	True
J48	confidenceFactor	(0.1;0.9, 9)	0.23
	minNumObj	(1;10, 10)	1
	numFolds	(1;10, 10)	3
	seed	(1;100, 20)	1
REPTree	minNum	(1;10, 10)	2
	minVarianceProp	(0.001)	0.001
	numFolds	(1;10, 10)	3
	maxDepth	(-5;30, 36)	22
	seed	(1;30, 30)	1
BFTree	minNumObj	(1;10, 10)	2
	numFoldsPruning	(1;15, 15)	5
	seed	(1;100, 20)	80
	sizePer	(0.1; 1.0, 10)	1.0
SimpleCart	minNumObj	(1;10, 10)	2.0
	numFoldsPruning	(1;30, 30)	5
	seed	(1;20, 20)	10
	sizePer	(0.1; 1.0, 10)	1.0
ADTree	numBoostingIteration	(100;1000, 10)	1000
	randomSeed	(0;10, 11)	0
	searchPath	(-3;0, 4)	0
LADTree	numBoostingIterations	(10;100, 10)	83
Random Tree	KValue	(1;100, 100)	48
	maxDepth	(0)	0
	minNum	(1;10, 10)	1.0
	numFolds	(0;100, 20)	0
	seed	(1;10, 10)	1

A weak point of the approach is that it takes long time to examine ranges of attributes. Another infirm factor of this is difficult to decide how big step be suitable for considerations.

Table 2. Results of improvements.

Algorithm	ACC	ACC in [3]	Difference
RandomForest	95.83	94.68	1.15
LMT	94.11	93.74	0.37
FT	93.94	93.24	0.70
J48graft	93.65	93.28	0.37
NBTree	93.20	93.20	0.00
J48	93.22	93.00	0.22
REPTree	92.89	92.81	0.08
BFTree	92.89	92.74	0.15
SimpleCart	92.76	92.26	0.50
ADTree	95.33	92.13	3.20
LADTree	94.76	92.09	2.67
RandomTree	92.78	91.05	1.73

Results in Table 2 show that our proposed methods obtained competitive results in comparison with that in [3]. Except for NBTree, all algorithms' accuracy are higher than that in [3] as showed in the last column. The average of accuracy improvements is 0.93%.

The improvements of J48, ADTree, and SimpleCART accuracy in term of correctly classified e-mail are showed in Table 3.

Table 3. Results of improvements.

Algorithm	Correctly classified instances	
	this paper	in [2]
J48	4289 (93.22%)	4268 (92.76%)
ADTree	4386 (95.33%)	4183 (90.92%)
SimpleCART	4268 (92.76%)	4262 (92.63%)

C. *Experiments For Feature Selection*

Various search and evaluation methods are combined to obtain best ACC for each algorithm as described in Table 4. The best and the second improvements are produced by RandomTree (2.06%) and RandomForest (1.53%), respectively. The datasets used in the experiments are named by a *d* and number of attributes selected (as in the 2nd column). For example, RandomForest is run on 56 attributes. Detailed information such as Attribute Evaluator and Search Method of all 8 datasets could be downloaded from https://goo.gl/V2hL3L.

Table 4. Best ACC for each algorithm based on feature selection

Algorithms	Dataset	New ACC	ACC in [3]	Difference
RandomForest	d56	96.21	94.68	1.53
LMT	d26	94.36	93.74	0.62
FT	d54	94.53	93.24	1.29
J48graft	d24	94.19	93.28	0.91
NBTree	d17	94.14	93.20	0.94
J48	d24	93.92	92.98	0.94
REPTree	d31	93.20	92.81	0.39
BFTree	d24	93.82	92.74	1.08
SimpleCart	d37	93.84	92.26	1.58
ADTree	d17	92.88	92.13	0.75
LADTree	d26	93.33	92.09	1.24
RandomTree	d20	93.11	91.05	2.06

The average of accuracy improvements is 1.11% as calculated from the last column.

D. *Experiments For Classifiers Combination*

Improvements of accuracy for 14 Stacking-based classifiers algorithms are described in Table 5. The highest accuracy is 96%, 1.32 higher than that of original one in [3].

Table 5. Stacking-based classifiers combination

No	**Classifier**	**ACC**
1	RandomForest+DMNBText+IBk+PART	96.00
2	RandomForest+Ibk+JRip	95.96
3	RandomForest+IBk+BayesNet+JRip	95.96
4	RandomForest+KStar+PART	95.78
5	RandomForest+IBk +DTNB	95.72
6	RandomForest+DMNBText+IBk+JRip	95.63
7	RandomForest+DMNBText+KStar+PART	95.50
8	RandomForest+NavieBayes+JRip	95.48
9	RandomForest+NavieBayes+Ibk+JRip	95.44
10	RandomForest+BayesNet+OneR	95.30
11	RandomForest + REPTree+ JRip	95.37
12	ADTree+RandomForest+ PART	95.37
13	SimpleCart+RandomForest+ JRip	95.37
14	RandomForest+ REPTree+ PART	95.33

Different results with some performance measurements based on ensemble algorithms Voting and Stacking could be downloaded from https://goo.gl/x9LEIk.

Like fine-tuning technique, a weak point of this approach is that it takes long time to examine various combinations of classifiers.

IV. Conclusion

We have introduced three tree-based approaches for spam filtering. We have performed the experiments in order to improve the classification accuracy of 12 algorithms. All experiments is to determine whether a particular e-mail is spam or not with the help of a data mining tool WEKA.

Experiment results show that the proposed methods can reach better accuracy classification in comparisons with some recent works in [2], [3]. The second approach, feature selection, is showed to be the best one in terms of both running time and accuracy improvement.

In future works, we seek to extend the model to other data representations and apply it to a wide range of spam types, such as Blog spam, SMS spam and Web spam. We predict that better results would be obtain if a more powerful computer system is used for experiments. This will be a good research direction of us.

Acknowledgment

This work was partially supported by Thai Nguyen University for university's research, code number DH2017-TN01-03.

References

[1] T. Shcherbakova, M. Vergelis and N. Dem, "Spam in July 2014," [Online]. Available: https://securelist.com. [Accessed 3 June 2017].

[2] I. Muhammad, A. Malik Muneeb, A. Mushtaq and K. Faisal, "Study on the Effectiveness of Spam Detection Technologies," *International Journal of Information Technology and Computer Science*, vol. 1, pp. 11-21, 2016.

[3] K. A. Sharma and S. Sahni , "A Comparative Study of Classification Algorithms for Spam Data Analysis," *International Journal a'xzon*, vol. 3, no. 5, pp. 1890-1895, 2011.

[4] E. Akçetin and U. Çelik, "The performance benchmark of decision tree algorithms for spam e-mail detection," *Journal of Internet applications and management*, vol. 5, no. 2, pp. 43-56, 2014.

[5] "UCI Machine Learning Repository," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase. [Accessed 4 June 2017].

[6] S. A. Elhamayed, "Classifying Datasets Using Some Different Classification Methods," *International Journal of Engineering and Technical Research (IJETR)*, vol. 5, no. 2, pp. 148-154, 2016.

[7] N. V. Truong, P. D. Lam and V. D. Quang, "Some improvements of selection algorithms for spam email filtering," *Journal of Science and Technology, Thai Nguyen University, Viet Nam*, vol. 151, no. 6, pp. 85-91, 2016.

[8] "Parameters Optimization in Weka," [Online]. Available: https://weka.wikispaces.com/Optimizing+parameters. [Accessed 4 June 2017].

[9] B. Jason, "How to Use Ensemble Machine Learning Algorithms in Weka," [Online]. Available: http://machinelearningmastery.com. [Accessed 6 June 2017].

[10] B. Jason, "Improve Machine Learning Results with Boosting, Bagging and Blending Ensemble Methods in Weka," [Online]. Available: http://machinelearningmastery.com/improve-machine-learning-results-with-boosting-bagging-and-blending-ensemble-methods-in-weka/. [Accessed 8 June 2017].

BIOGRAPHY

Nguyen Van Truong is a lecturer in the Faculty of Mathematics at Thai Nguyen University of Education, from where he received a Bachelor of Mathematics and Informatics in 2000. He finished his master course on Computer science at Vietnamese National University in 2003. He is currently a PhD student at Institute of Information Technology (IOIT), Vietnamese Academy of Science and Technology (VAST). He has taught a wide variety of courses for UG students and guided several projects. He has published several papers in National Journals & International Conferences. His research interests are embedded systems and artificial immune systems.

Doan Minh Thai is a lecturer in the Faculty of Mathematics at Thai Nguyen University of Education, from where she received a Bachelor of Informatics in 2004. She finished her master course on Computer science at Thai Nguyen University of Information and Communication Technology in 2007. She teaches Discrete Math, High-level programming language, Design and Analysis of Algorithms, etc.

Le Bich Lien is a lecturer in the Faculty of Mathematics at Thai Nguyen University of Education, from where she received a Bachelor of Informatics in 2004. She finished her master course on Computer science at Thai Nguyen University of Information and Communication Technology in 2007. She teaches Web Design and Programming, High level programming language, etc.

Nguyen Thi Thu is a Bachelor of Informatics Pedagogy at Thai Nguyen University of Education. Besides, she is also a senior student in mathematics at this university. She is interested in Design and Analysis of Algorithms