Email Header Feature Extraction using Adaptive and Collaborative approach for Email Classification

Amandeep Singh Rajput, J S Sohal, Vijay Athavale

Abstract: ***Email Header is footprint of an Email that can be used to examine an Email as HAM or SPAM. Email classification in this research is done on the basis of header features thus by keeping the content privacy of the sender intact [1]. Header features are , email header fields like sender, to, cc, bcc, subject. This research tries to improve the accuracy of the classification by extracting more number of header features. Email Subject is further deeply examined for objectionable keywords for rule matching and rule generation. In our study, we implement an adaptive and collaborative approach by using machine learning and cluster computing for fast classification of Emails as SPAM or HAM. Adaptive approach is to generate new rules for classification and cluster approach is to use parallel computing power for increasing computing speed. New rules are only generated if features extracted from email header do not match the existing rules. Spam Assassin [2][3] is the main dataset used for testing. Collaborative approach creates a parallel environment where multiple antispam methods and divided test corpora are used as input. The false positive and false negative percentage are recorded and accuracy is calculated. Weka Data Mining Software is used to apply the anti-spam methods (available at http://www. cs.waikato.ac.nz /~ml/weka/) [4]***

Keywords: Classification, Features, Ham, Spam, Machine Learning, Corpora, Parallel Environment, CrossValidation, Weka.

I. INTRODUCTION

SPAM are still a major concern as average daily spam volume is about 311.24 Billion for the month Dec-18, which counts to 85% of monthly emails which are SPAM [5]. Electronic mail header is an effective data to examine an email as SPAM. An email header includes fields like From, To, Subject, CC (Carbon Copy), BCC (Blind Carbon Copy). Blind Carbon Copy if used, then very likely it can be classified as SPAM, as. Spammers want to spread SPAM as much as possible and BCC can be used for this purpose. Spammers use BCC to reach out to more number of email users and also there is no restriction on the number of emails address that can be used in the BCC field [6]. Email Subject is most important among these features. Subject can further be tokenized, into words, phrases, character. The frequency of tokenized words or phrases help in classify the email as HAM or SPAM.SPAM mail can be identified in two ways, first is at the origin and second is at the destination.

There are some black listed sources on the internet that can be considered as origin of SPAM and can instantly blocked. More challenging task is finding an SPAM email when it lands at destination server. These mails need exhaustive analysis to classify them as SPAM or HAM. The main objective of this research is to decrease the percentage of false positive and false negative during email classification. Another objective is to increase the computing power and speed to carry out this exhaustive analysis using cluster computing. The various steps involved in email header features extraction are listed as follows.

- Collect email headers from spamassian, lingspam, personal mail.
- Spamassian data is directly used.
- Extract features from, to, cc, bcc, subject from personal mail.
- Tokenize the email subject.
- Extract and retain useful tokens that are useful for classification.
- Tokens that are not useful are removed before they can be used for classification purpose.
- Frequency of each retained token is calculated.

Unnecessary email header features if reduced, increases the classification speed [7]. The classification here is done on a spam scale percentage rather than classifying emails into two strict classes i.e ham or spam. If spam scale is 100% then email in question is a spam, if the scale is 75% then it is towards spam, if scale is 50% then it can fall into either class, if the scale is 25% it is towards ham and if scale is 0% then it is a ham.

This paper is divided in four sections. The first section explains about the email header features. The second section focuses on the related work and literature review in email header feature extraction and classification. The third section highlights the adaptive and collaborative approach being used in this research along with the machine learning techniques applied. The forth section implements feature extraction and classification using collaborative approach. The fifth section is the result compilation and analysis.

II. SECTION I - EMAIL HEADER FEATURES

Email as a whole is a complex entity, where various emails protocols, time-stamps, sender server, receiver server get involved. Thus we are focusing only on email header to have focused area of study.

Revised Manuscript Received on May 05, 2019.

Amandeep Singh Rajput, Computer Science Department, GGI Khanna, Affiliated to IGK Punjab Technical University, Kapurthala
J S Sohal, Ludhiana College of Engineering & Technology, Ludhiana
Vijay Athavale, ABESEC, Ghaziabad

Header Features like From, To, Carbon Copy, Blind Carbon Copy, Subject are used as input to email classification method. This research tries to improve the accuracy of the classification by extracting more number of header features. Though email content and attachments can reveil much more information than the email header features but on the other hand it compromises the privacy of the user, which in many countries is against the law. The next section shows that email header alone can give 80% accuracy in classifying emails. Email header constitutes of source of email, reciepents of email, just for information receipients i.e. cc and bcc, subject of email. Our interest in this study is the subject of the email. One important thing regarding subject of the email is that it is the only part of the email that is available to the receipient which can be seen before opening the email. The content of the mail is only available to the user once the user decides to open the email. To lure the user spammers make the subject line attractive so that users are dragged into the decision of opening the email. Another aspect is that email cannot be sent without a subject, on the other hand content of the email can be skiped by the sender just by adding attachements. Email header can be considered as the identification of an email or a footprint of an email which almost reveils the nature of the email. Thus email header, more so the subject of the email is a useful email header feature that is of prime importance in study.

III. SECTION II – LITERATURE REVIEW

Email classification on the basis of header features keeps the content privacy intact. intact[1]. It is already demonstrated that the accuracy of classification still gives resonable results as compared to content based classifiers [1] . Many researchers use only header based classifiers. If considering only non-content base classification, then header based spam filtering gives good results [6]. Ye [8] proposed a model based on Support Vector Machine (SVM) to discriminate spam messages depends on mail header features. For each email header a feature was extracted by [7]. Email header is a useful part of email that can disclose many features in classifying an email. Email subject can be tokenized and frequency of dictionary words leading to spam can be counted. Email header features were divided into dictionary words, HTML tags and words with numerals by R. Shams and R. E. Mercer [9]. They used learning algorithms like Random Forest , Naive Bayes Classifier, Support Vector Machine, AdabbostM1 and Bagging. Random Forest was choosen for study by R. Shams and R. E. Mercer [9] since it is widely used anti spam method, runs efficiently on large data sets and the learning speed or the model training speed is very high. Graham [10] pointed out that only by extracting and examining words from email header 79.7% of spam emails in a dataset can be detected which count for only 1.2% false positives, which in turn is quite remarkable considering only the header feature. The information provided by email header is quite important. This is confirmed in their study by Nizamani, Memon, Glasdam[11]. They [11] concluded that email header feature are more important than the classification method being used. Though there is no single feature and classification method that can be zeroed down for perfect classification, Youn and McLeod [12] showed that WEKA implemented J48 classification method gives better results than Naïve Bayes, SVM and Neural Network classifiers. Younand McLeod[12] also pointed out that it is important to select and extract the relavent feature that point to spam mails. Campos, Verdejo and Teodoro[13] focused on technique which is based on the information stored in e-mail header and not on the email body to keep the privacy of the user intact. Lai, Chen and Laih[14] identified a collaborative framework on spam rule generation, exchange and management. The collaborative approach by [14] exchanged spam filter rules between classifiers with the help of XMLfiles. Feature extraction was used for contents of email, files attached and images by [15]Blanzieri and Bryl. Though email header feature extraction was not used by [15]Blanzieri and Bryl. Zhou, Yao and Luo [16], instead of classifying emails into just two categories used three calegories to classify mails, i.e. spam, ham and undecided. The undecided mails need more analysis before identifying as spam or ham. In our study during this research we have taken one step further to have five categories, i.e. spam, towards spam, ham, towards ham and undecided. Classifying email data directly into two classes i.e. ham or spam increases the possiblity of false positive and false negative, which is not good for any classifier. Data from three different sources and ten fold cross-validation method was used by Campos, Verdejo and Teodoro [17] to classifiy email data. The author used latest data till year 2008, no personal mail data was used. In our research we try to use considerable amout of personal email data, just to have a wider scope of this research. Personal emails of about 0.5M in number used in this study. These mails are made public by FERC at https://www.cs.cmu.edu/ ~./enron/ . In any case pesonal mails, more than ten thousand in number were used by Alsmadi and Alhami[18] for analysis. During email feature extraction stemming technique was used by Alsmadiand Alhami[18] to phase out irralevent words like to, you, I, am, pm etc.

IV. SECTION III - ADAPTIVE AND COLLABORATIVE APPROACH

Standalone machines have definitely limited computing power and speed as compared to parallel computing macines. With limited computing power and speed, standalone servers do not perform efficiently [19]. It is more evident when large amount of data is to be processed repeatedly. Spam classification is an area where no single classification algorithm and training dataset is sufficient for perfect classification. This research takes a view on handling dynamic nature of spam mails with adaptive aproach i.e. generating new rules and the collaborative approach is used to applying multiple classification algorithms and for extracting header features. Spammers and anti spammers are always in a tussle to be ahead of other. Therefore it is always necesssry to generate new anti spam rules. In this research the focus is on adapatability and speed at low cost.

V. SECTION IV - FEATURE EXTRACTION IMPLEMENTION

The datasets used here are spambase available at https://archive.ics.uci.edu/ml/ datasets/Spambase[20], and personal mail data. The dataset other than personal mails are already feature extracted and need not be reprocessed. The personal mails are available in raw format and hence needs header feature extraction. Personal mail at https:// www.cs.cmu.edu /~./enron/[21], which are large in number, 0.5M, are feature extracted first and then normalized and then fed to weka server for classificatiion. The subject words in email header can be analysed to see if all letters are capital, if that is the case it is likely that it is a spam as spammers try to highlight or attarct attention by putting every letter in capital. Also cleaverly written words like Money written as M0ney, mo.ney, m o n e y, mooney, M O N E Y etc. are some of the tricks used by spammers and are taken care of during preprocessing. During the preprocessing stage, a python script is used to segrigate such email as spam.

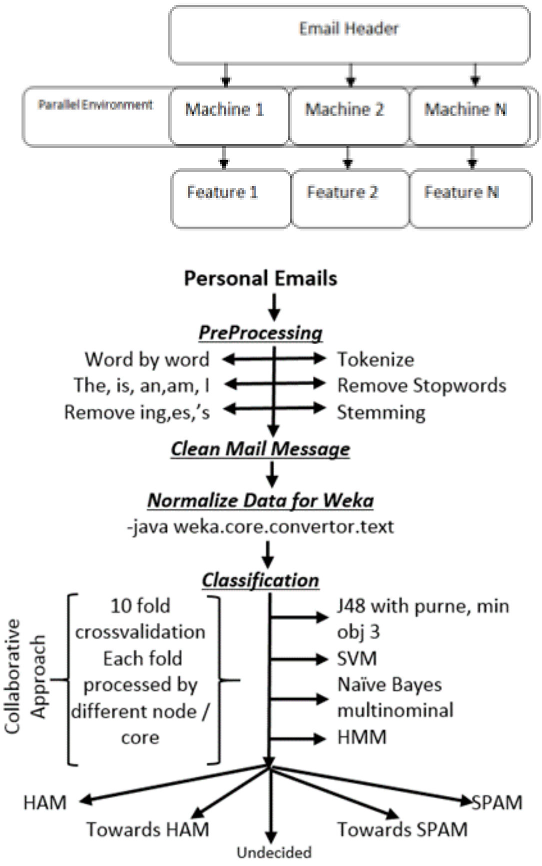
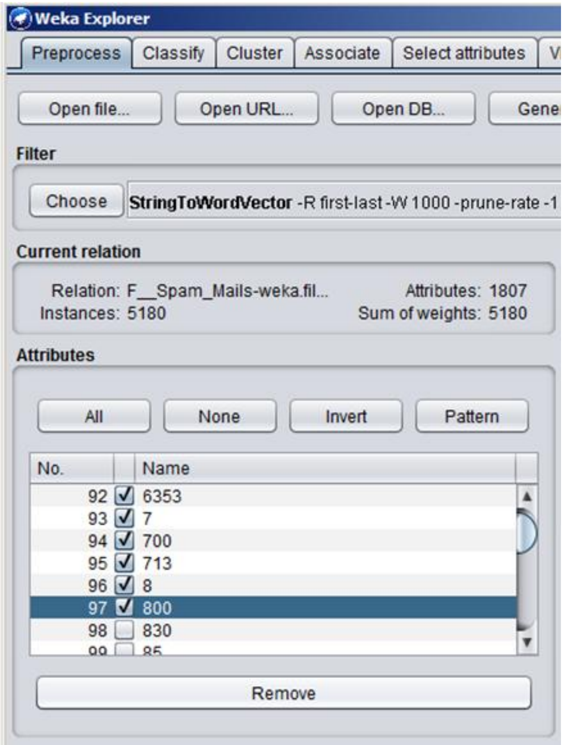


Fig. 1 Preprocessing and Classification Process

Personal email data needs preprocessing in the form of tokenizing the email subject word by word and each word treated as a header feature. To create personal email arff file, first emails are downloaded by email backup tool "Got Your Back" (GYB) available at https://github.com/jay0lee/got-your-back/ wiki [22] and then subject alone is extracted from each email and a text file is created. To tokenize subject word by word, wekaStringToWordVector filter is applied. Numeric values like date, serial number or any other kind of numbers are excluded from the featureset by using the remove attribute feature of weka. The experiment divided each dataset into two futher sets as , less header features and all header features. Then the accuracy of each classifier noted down. The experiment results are shown in result section.

VI. SECTION V – EMAIL CLASSIFICATION IMPLEMENTATION

A two node Linux cluster/parallel environment is created on Linux with openMosix. Weka data mining tool is to be used to apply machine learning and non machine learning methods on the corpus. Weka server is installed along with the java8 on this cluster. Further experiments are carried under this environment.

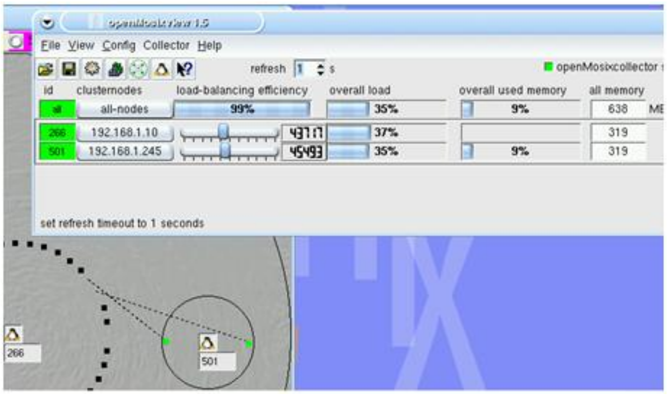


StringToWordVector -R first-last -W 1000 -prune-rate -1

Relation: F__Spam_Mails-weka.fil... Attributes: 1807
Instances: 5180 Sum of weights: 5180

No.		Name
92	☑	6353
93	☑	7
94	☑	700
95	☑	713
96	☑	8
97	☑	800
98	□	830
99	□	85

Fig. 2 Attribute Selection and Removal



id	clusternodes	load-balancing efficiency	overall load	overall used memory	all memory
all	all-nodes	99%	35%	9%	638
266	192.168.1.10	4311	37%		319
501	192.168.1.245	4549	35%	9%	319

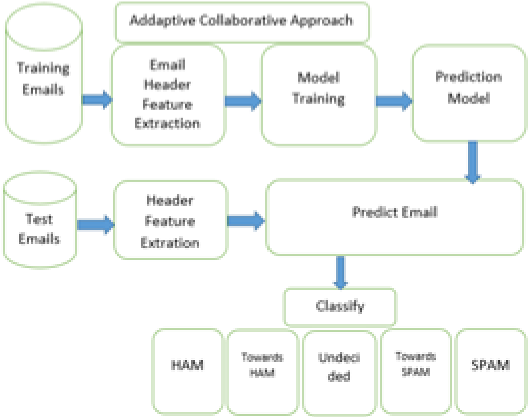
set refresh timeout to 1 seconds

Fig. 3 Two Node Cluster for Testing

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Some steps used for setting parallel environment.

```
# ifconfig eth0 up ( Lan cards setup )
# ifconfig eth0 192.168.1.10 (assign IP)
# route add -net 0.0.0.0 gw 192.168.1.1
( setting the route and matric )
# /etc/init.d/openmosix restart
# omdiscd -i eth0 ( auto detection client)
# openmosixview& ( start openmosixview )
```



1) Support Vector Machines (SVM) uses supervised learning techniques for email classification. SVM separates data in multi-dimensional plane so that there is isolation plane which separates the classes.

2) The Naïve Bayes classifier is a complex algorithm that treats data items independent of each other. The Naïve Bayes classifier uses conditional probability and useful for huge datasets. Being simple and easy to understand and implement it is used by many researchers. It also know to outperform many other classifiers.

3) J48 is weka's modified version of C45 decision tree algorithm. It is a predictive model that does not treat data item independently as in case of Naïve Bayes. The internal nodes of a tree is an indication of different attributes and the branches provides us with the possible values of the attributes. The leaf nodes tell us about the class the data belongs to.

4) Hidden Markov Model is a classification model based on probabilities to predict hidden variables from a given set of observations. Here we are predicting class from header features.

W is word in a email header, which is extracted from subject. F is the frquency of the word W.

$$\text{spam}[W] = \frac{F(\text{spam}[W])}{F(\text{spam}[W]) + F(\text{ham}[W])}$$

Naïve Bayes Approach to classify as SPAM or HAM.

Spam[W] gives a value that indicates if the word W is spam word or not.

Spam[M] is claculated by averaging spam[W] for all words in that message.

The value of spam[M] decides the class of the message.

Table. 1 Decision Factor Table

Spam[M] value					
Decision Factor	< 25	25-49	50	51-75	>75
Personal data classification	Ham	Ham like	Un-decided	Spam like	spam

Weka Server utilizes each node processor power in a parallel environment or multiple core capabilities of signle processor for classification tasks. Weka experimenter is also available for load sharing on remote computers. Weka experimenter allows remote experiments to be spread across multiple hosts, thus by creating a parallel environment i.e. applying a collaborative approach. Command used in weka CLI is *:-java -Djava.awt.headless=true weka.RunWekaServer -host 192.168.1.10 -port 8085 -master 192.168.1.10:8085 -slots 4.*

VII. SECTION IV - FEATURE EXTRACTION IMPLEMENTION

1. Header Features and Accuracy:- In case of Spambase data the number of header features are directly propotional to the accuracy of the classifier. The experiment showed the same trend of increased accuracy with increased email header features irrespective of the classifier used, and the data set used.

Table. 2 Relation of Header Features to Accuracy

Corpus	Number of Instances	Classifier	Number of Header Features	Accuracy
Personal Emails	5180	NBayes	1572	92.77 %
			1692	92.81 %
		J48	1572	92.91 %
			1692	93.66 %
		SVM	1572	96.73 %
			1692	96.91 %
		HMM	1572	70.88 %
			1692	70.88 %
Spambase	4601	J48	58	92.97 %
			47	92.71 %
		Nbayes	58	89.80 %
			47	88.45 %
		SVM	58	90.43 %
			47	89.00 %
		HMM	58	39.40 %
			47	39.40 %

2. Personal data classification:- For each message a spam factor value is calculated as per the formula discusses in Section V. The message file is first split into two files, i.e file with spam messges and file with ham messages. Next the frequency of each word in each file is calculated with the help of wekaString To WordVector feature and output Word Counts setting of this feature. A python script is used to calculate spam[M] value of each message. The python script is tested and developed on spyder version 3.2.8 and python version 3.6. The evaluation process for one typical message is shown below.

Message:- “shop and test this homepage for all wonderful photographs and videos for free”. This message is actually a spam. Our classification method finds the frequency of main key words like ‘shop’, ‘test’, ‘homepage’, ‘wonderful’, ‘photographs’, ‘videos’, and ‘free’. Other words in message like ‘and’, ’this’, ‘for’, ’all’ are excluded from analysis. As these words do not contribute in decision making process. To find if through our classification method, the given message is spam, spam like , ham , ham like or undecided we find spam factor of the message as in the table below.

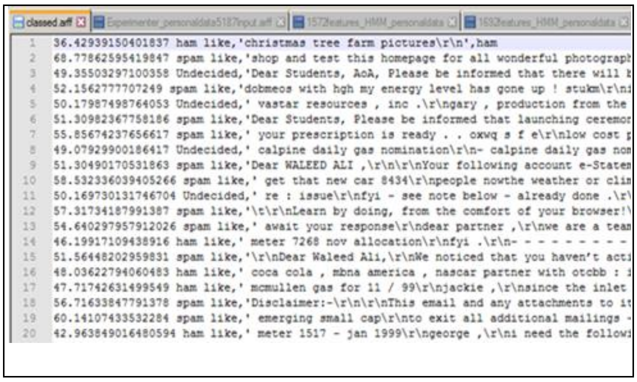
Table. 3 Decision Process for a message

word	Freq in Ham message	Freq in Spam messages	Spam[w]
Shop	67	178	0.72
Test	155	175	0.53
homepage	4	4	0.50
wonderful	10	9	0.47
Photographs	0	2	1.0
videos	2	27	0.93
free	301	368	0.55
Spam[M] = Avg(spam[w]) x 100 =			***67***

The message analysed in Table-3 gives a spam factor of 67, which is between 51 to 75 and as per our decision factor table, the message should be classified as “Spam Like” and not as spam. This indicates that the messages which are not in Ham or Spam class needs further analysis. The resutls of the complete personal data file is summarised in the table below.

Table. 2 Classification in more than one class

Spam[M] value						
		< 25	25-49	50	51-75	>75
	Personal data	Ham	Ham like	Un-decided	Spam like	spam
1	Actual	3672	--	--	--	1508
2	Classified	1098	2113	2	1031	936


classed.arff
Experimenter_personaldata5187input.arff
1572features_HMM_personaldata
1692features_HMM_personaldata

```
36.42939150401837 ham like,'christmas tree farm pictures\r\n',ham
68.77862595419847 spam like,'shop and test this homepage for all wonderful photograph
49.35503297100358 Undecided,'Dear Students, AoA, Please be informed that there will b
52.1562777707249 spam like,'dobmeos with hgh my energy level has gone up ! stukm\r\ni
50.17987498764053 Undecided,' vastar resources , inc .\r\ngary , production from the
51.30982367758186 spam like,'Dear Students, Please be informed that launching ceremon
55.85674237656617 spam like,' your prescription is ready . . oxwq s f e\r\nlow cost p
49.07929900186417 Undecided,' calpine daily gas nomination\r\n- calpine daily gas nom
51.30490170531863 spam like,'Dear WALEED ALI ,\r\n\r\nYour following account e-Statem
58.53233603940526б spam like,' get that new car 8434\r\npeople nowthe weather or clim
50.16973013174670 4 Undecided,' re : issue\r\nfyi - see note below - already done .\r\
57.3173418799l387 spam like,'\t\r\nLearn by doing, from the comfort of your browser!\
54.64029795791202б spam like,' await your response\r\ndear partner ,\r\nwe are a team
46.1991710943891б ham like,' meter 7268 nov allocation\r\nfyi .\r\n- - - - - - - - -
51.5644820295983l spam like,'\r\nDear Waleed Ali,\r\nWe noticed that you haven't acti
48.03622794060483 ham like,' coca cola , mbna america , nascar partner with otcbb : i
47.71742631499549 ham like,' mcmullen gas for 11 / 99\r\njackie ,\r\nsince the inlet
56.71633847791378 spam like,'Disclaimer:-\r\n\r\nThis email and any attachments to it
60.14107433532284 spam like,' emerging small cap\r\nto exit all additional mailings -
42.96384901648059 4 ham like,' meter 1517 - jan 1999\r\ngeorge ,\r\ni need the followi
```

Fig. 4 Python Script output, message classification, ham, hamlike, spam, spamlike

3. Parallel environment

The below table shows the comparison of execution time of four machine learning algorithms used in this research. The execution time for standalone system is automatically recorded during the classification process, which can be seen in figure-7 and figure-8. The execution time for server runs are calculated from server screen shown in figure-6. The complete experiment uses 5180 instances and 1692 header features of email subject. In the line graph in figure-5 it is clear that there is a decline in execution speed in case of SVM, HMM, and Naïve Bayes algorithm. This is because the due to small dataset, the overheads of parallel environment like massage passing and load balancing reduces performance. In the case of J48 algorithm the size of tree large, thus we can see a little performance improvement in this case.

Table. 3 Time comparison in Parallel environment Classification Execution

Classification Method	Data Set	Standalone Time in secs	Parallel Environment Time in Secs
SVM	Personal Data 5180 Instance 1692 features	8.3	48.01
J48		152.26	148.12
HMM		0.11	15.18
NaiveBayes		3.48	34.71

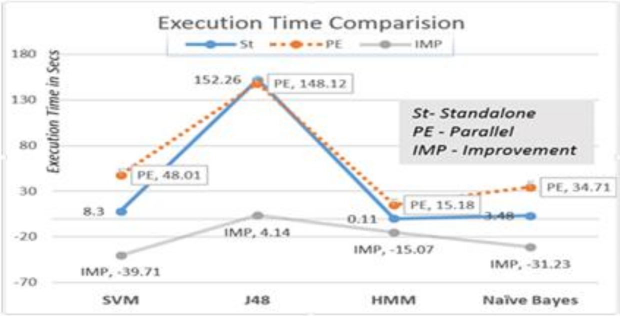
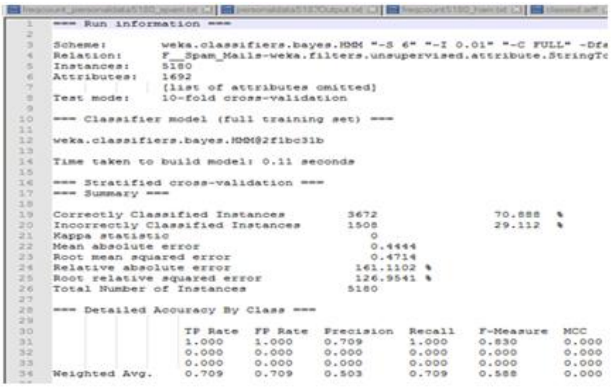


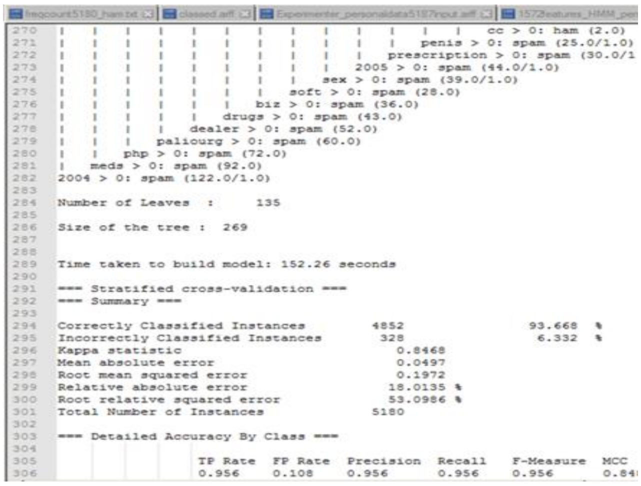
Fig. 5 Execution Time Comparision Standalone vs Parallel Env

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication


freqcount_personaldata5180_spam.txt
personaldata5187Output.txt
freqcount5180_ham.txt
classed.arff

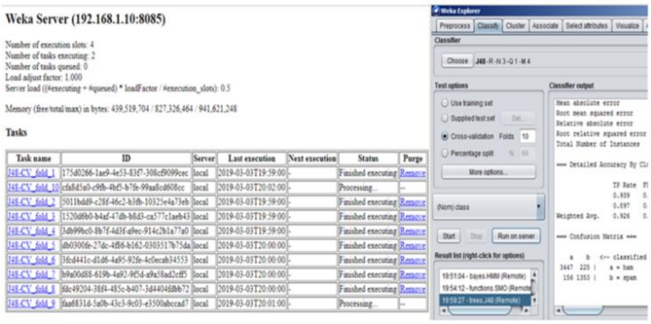
```
=== Run information ===

Scheme:       weka.classifiers.bayes.HMM "-S 6" "-I 0.01" "-C FULL" -Dfa
Relation:     F__Spam_Mails-weka.filters.unsupervised.attribute.StringTc
Instances:    5180
Attributes:   1692
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

weka.classifiers.bayes.HMM@2f1bc31b

Time taken to build model: 0.11 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3672               70.888 %
Incorrectly Classified Instances      1508               29.112 %
Kappa statistic                          0
Mean absolute error                      0.4444
Root mean squared error                  0.4714
Relative absolute error                161.1102 %
Root relative squared error            126.9541 %
Total Number of Instances             5180

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 1.000    1.000    0.709      1.000    0.830      0.000
                 0.000    0.000    0.000      0.000    0.000      0.000
                 0.000    0.000    0.000      0.000    0.000      0.000
Weighted Avg.    0.709    0.709    0.503      0.709    0.588      0.000
```

Fig. 6 HMM screenshot for standalone execution for personal dataset



```
|   |   |   |   |   |   |   |   |   |   |   |   |   cc > 0: ham (2.0)
|   |   |   |   |   |   |   |   |   |   |   penis > 0: spam (25.0/1.0)
|   |   |   |   |   |   |   |   |   |   prescription > 0: spam (30.0/1
|   |   |   |   |   |   |   |   |   2005 > 0: spam (44.0/1.0)
|   |   |   |   |   |   |   |   sex > 0: spam (39.0/1.0)
|   |   |   |   |   |   |   soft > 0: spam (28.0)
|   |   |   |   |   |   biz > 0: spam (36.0)
|   |   |   |   |   drugs > 0: spam (43.0)
|   |   |   |   dealer > 0: spam (52.0)
|   |   |   paliourg > 0: spam (60.0)
|   |   php > 0: spam (72.0)
|   meds > 0: spam (92.0)
2004 > 0: spam (122.0/1.0)

Number of Leaves  :     135

Size of the tree :  269


Time taken to build model: 152.26 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4852               93.668 %
Incorrectly Classified Instances       328                6.332 %
Kappa statistic                          0.8468
Mean absolute error                      0.0497
Root mean squared error                  0.1972
Relative absolute error                 18.0135 %
Root relative squared error             53.0986 %
Total Number of Instances             5180

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.956    0.108    0.956      0.956    0.956      0.84
```

Fig. 7 J48 screenshot for standalone execution for personal dataset


Weka Server (192.168.1.10:8085)

Number of execution slots: 4
Number of tasks executing: 2
Number of tasks queued: 0
Load adjust factor: 1.000
Server load ((#executing + #queued) * loadFactor / #execution_slots): 0.5

Memory (free/total/max) in bytes: 439,519,704 / 827,326,464 / 941,621,248

Tasks

Task name	ID	Server	Last execution	Next execution	Status	Purge
J48-CV_fold_1	175d0266-1ae9-4e53-83f7-308cf9099cec	local	2019-03-03T19:59:00	-	Finished executing	Remove
J48-CV_fold_10	cfa8d5a0-c9fb-4bf5-b7fe-99aa8cd608cc	local	2019-03-03T20:02:00	-	Processing...	--
J48-CV_fold_2	5011bdd9-c28f-46c2-b3fb-10325e4a73eb	local	2019-03-03T19:59:00	-	Finished executing	Remove
J48-CV_fold_3	1520d6b0-b4af-47db-b8d3-ca577c1aeb43	local	2019-03-03T19:59:00	-	Finished executing	Remove
J48-CV_fold_4	3db99bc0-8b7f-4d3f-a9ec-914c2b1a77a0	local	2019-03-03T19:59:00	-	Finished executing	Remove
J48-CV_fold_5	db0300fe-27dc-4f86-b162-0303517b75da	local	2019-03-03T20:00:00	-	Finished executing	Remove
J48-CV_fold_6	3fcd441c-d1d6-4a95-92fe-4c0ecab34553	local	2019-03-03T20:00:00	-	Finished executing	Remove
J48-CV_fold_7	b9a00d88-619b-4a92-9f5d-a9a58ad2cff5	local	2019-03-03T20:00:00	-	Finished executing	Remove
J48-CV_fold_8	fdc49204-38f4-485c-b407-3d44046fdbb72	local	2019-03-03T20:00:00	-	Finished executing	Remove
J48-CV_fold_9	faa6831d-5a0b-43c3-9c03-e3500abccad7	local	2019-03-03T20:01:00	-	Processing...	--

J48 -R -N 3 -Q 1 -M 4

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

19:51:04 - bayes.HMM (Remote)
19:54:12 - functions.SMO (Remote)
19:59:27 - trees.J48 (Remote)

```
Mean absolute error
Root mean squared error
Relative absolute error
Root relative squared error
Total Number of Instances

=== Detailed Accuracy By Cl

                 TP Rate  F
                 0.939    0.
                 0.897    0.
Weighted Avg.    0.926    0.

=== Confusion Matrix ===

    a    b   <-- classified
 3447  225 |    a = ham
  156 1353 |    b = spam
```

Fig. 8 Classification Run of Weka Server

VIII. SECTION VI: CONCLUSION

1. A message classified through machine learning algorithms need not to be completely accurate and needs more analysis as the classification largely depends on the training dataset used, algorithms used and the number of header features used. To find out the best combination of the machine learning algorithms for classification remains as the part of future scope of this research.

2. Parallel environments only help if there is a huge dataset. For small data sets, for experiment purpose, parallel environments do not show any improvement as the parallel setup is complex and message passing and load balancing tools overburden the small configuration systems as used in this research. In the case of cross validation ten-folds on weka server that takes advantage parallel environment of executing each fold on each core of multicore system only showed marginal improvement in J48 tree algo where the size of tree is very large.

REFERENCES

1. ZhenhaiDuan a, KartikGopalan b, Xin Yuan a, "An empirical study of behavioral characteristics of spammers: Findings," Computer Communications, vol. 34, p. 1764–1776, 2011.
2. The Apache Software Foundation, "http://spamassassin.apache.org/downloads.cgi," 2015. [Online]. Available: http://www-us.apache.org/dist//spamassassin/source/Mail-SpamAssassin-3.4.1.tar.gz. [Accessed 20 August 2017].
3. GitHub, Inc, "SpamAssassin," 21 April 2016. [Online]. Available: https://github.com/dmitrynogin/SpamAssassin.git. [Accessed 20 August 2017].
4. U. o. Waikato, "Weka 3: Data Mining Software in Java," Machine Learning Group at the University of Waikato, 2017. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed 20 August 2017].
5. I. Cisco Systems, "email_rep.html," Cisco Systems, Inc., 30 Dec 2018. [Online]. Available: https://www.talosintelligence.com/reputation_center/email_rep. [Accessed 12 Jan 2019].
6. C. G. a. E. N. b. M. L. c. S. C. Yong Hua, "A scalable intelligent non-content-based spam-filtering framework," Expert Systems with Applications, vol. 37, no. 12, pp. 8557-8565, 2010.
7. Masoumeh Zareapoor, Seeja K. R, "Feature Extraction or Feature Selection for Text," I.J. Information Engineering and Electronic Business, vol. II, no. 7, pp. 60-65, March 2015.
8. M. Y, "A Spam Discrimination Based on Mail Header Feature and SVM," in Wireless Communications Networking and Mobile Computing WiCOM, 2008.
9. Rushdi Shams and Robert E. Mercer, "Classifying Spam Emails using Text and Readability Features," in IEEE 13th International Conference on Data Mining, Dallas, TX, 2013.
10. P. Graham, "A Plan for Spam," August 2002. [Online]. Available: http://paulgraham.com/spam.html. [Accessed 4 Feb 2019].
11. SarwatNizamani, NasrullahMemon, MathiesGlasdam,, "Detection of fraudulent emails by employing advanced feature abundance," Egyptian Informatics Journal (2014) 15, 169–174, vol. 15, no. 3, pp. 169-174, 2014.
12. SeongwookYoun, Dennis McLeod, "A Comparative Study for Email Classification," in Advances and Innovations in Systems, Computing Sciences and Software Engineering, 2007.
13. J. D.-V. P. G.-T. Francisco Salcedo-Campos, "Segmental parameterisation and statistical modelling of e-mail headers for spam detection," Information Sciences—Informatics and Computer Science, Intelligent Systems, Applications, vol. 195, no. 15, pp. 45-61, July 2012.
14. Gu-HsinLai, Chia-MeiChen, Chi-SungLaih, TsuhanChen, "A collaborative anti-spam system," Expert Systems with Applications, vol. 36, no. 3, pp. 6645-6653, April 2009.
15. Bryl, Enrico Blanzieri and Anton, "A Survey of Learning-Based Techniques of Email Spam Filtering," University of Trento, Italy, 2008.
16. Bing Zhou, Yiyu Yao, Jigang Luo, "A three way decision Approach to email spam filtering.," in Canadian Conference on AI, 2010.
17. Francisco Salcedo-Campos, JesúsDíaz-Verdejo, Pedro García-Teodoro, "Segmental parameterisation and statistical modelling of e-mail headers for spam detection," Information Sciences, vol. 195, pp. 45-61, 15 July 2012.
18. IzzatAlsmadi, IkdamAlhami, "Clustering and classification of email contents," Journal of King Saud University – Computer and Information Sciences, vol. 27, no. 1, pp. 46-57, 2015.
19. Gu-Hsin Lai a,*, Chia-Mei Chen a, Chi-Sung Laih b, Tsuhan Chen c, "A collaborative anti-spam system," Expert Systems with Applications, vol. 36, no. 3, pp. 6645-6653, 2009.
20. D. a. K. T. E. Dua, "Spambase Data Set," University of California, Irvine, School of Information and Computer Sciences, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Spambase. [Accessed 17 Feb 2019].

21. W. W. Cohen, "Enron Email Dataset," CALO Project, 7 May 2015. [Online]. Available: https://www.cs.cmu.edu/~./enron/enron_mail_20150507.tar.gz. [Accessed 17 Feb 2019].
22. Steve, "got-your-back," Github, 29 DEC 2018. [Online]. Available: https://github.com/jay0lee/got-your-back/wiki. [Accessed 15 FEB 2019].
23. Jessa dela Torre, Sabrina Lei Sioting, "Spam and All Things Salty: Spambot v2013," 03 12 2013. [Online]. Available: https://www.botconf.eu/wp-content/uploads/2013/12/03-JessadelaTorre-SpamBot-paper.pdf. [Accessed 20 August 2017].
24. C.-C. Wang, "Sender and Receiver Addresses as Cues for Anti-Spam," Journal of Research and Practice in Information Technology, vol. 36, no. 1, Feb 2004.
25. YinglianXie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, Ivan Osipkov, "Spamming Botnets: Signatures and Characteristics," ACM 978-1-60558-175-0/08/08, Seattle, Washington, USA., 2008.
26. D. R.-O. F. F.-R. a. J. R. M. N. Pérez-Díaz, "Boosting Accuracy of Classical Machine Learning Antispam Classifiers in Real Scenarios by Applying Rough Set Theory," Scientific Programming, Vols. 2016, 10 pages, 2016, no. Article ID 5945192, pp. 1-10, 2016.
27. Clotilde Lopes a, Paulo Cortez a,⇑, Pedro Sousa b, Miguel Rocha b, Miguel Rio c, "Symbiotic filtering for spam email detection," Expert Systems with Applications, vol. 38, no. 8, pp. 9365-9372, 2011.
28. Francisco Salcedo-Campos, JesúsDíaz-Verdejo, Pedro García-Teodoro, "Segmental parameterisation and statistical modelling of e-mail headers for spam detection," Information Sciences, vol. 195, pp. 45-61, 2012.
29. El-Sayed M. El-Alfy, Radwan E. Abdel-Aal, "Using GMDH-based networks for improved spam detection and email feature analysis," Applied Soft Computing, vol. 11, no. 1, p. 477–488, 2011.
30. "Ling-Spam datasets," CSMINING GROUP, 17 July 2000. [Online]. Available: Csmining.org. (2017). Ling-Spam datasets - Chttp://csmining.org/index.php/ling-spam-datasets.html. [Accessed 30 Sept. 2017].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication