Sementic Web Mining Methodlogies For Malwares Detection in Emails .

Dr. Ritu Bhargava
Lecturer, Computer Science
Sohpia Girls' College Ajmer, India
drritubhargava92@gmail.com

Abhishek Kumar
Assistant Professor ,CSE Dept.
Aryabhatta College of Engg. & Research Center, Ajmer India
Abhishekkmr812@gmail.com

Sweta Gupta
Research Scholar
MJRP University, Jaipur
swetagupta097@gmail.com

***Abstract*—.** Web mining is a process of online mining primarily in which all the web activities including their mining and security at the same time. A dataset for different incurring spam has been designed to get better accuracy for the work of web mining. The work is primarily based on semantic web mining concept along with the concept of detecting malicious attack with irrelevant data like spams, and other things in mails and social and professional networking sites. The work has been centered around providing an optimal pattern based on threshold curve and frequency of words in the designed dataset. In order to get better classifications Naïve Bayes classifier has been applied over the designed dataset .Weka tool has been used for analyzing the designed dataset. Performing semantic web mining with commuted concept of malwares detection in form of Spam etc. is the primary work. The work is based on a dataset which has been designed on the basis of spam on the mail. The repeated pattern of that has been monitored and detected with accuracy in results.

Keywords— web mining,naïve bayes,cross validation,attributes classifications.

I. INTRODUCTION

Semantic web mining is combined technique for two different methodologies, semantic web and data mining[1]. Web mining is primarily an web approach and needed to be implemented using world wide web. The researcher now a days using tools like Weka, Rapid miner etc for its implementation over the designed dataset as per requirements. The dimension of Semantic web mining is very much wide like web mining and web mining with security measures because all networking sites, mails, use web mining approach to classify the data ,mails, responses and malwares also.Sementic we b mining with threat protection considering different types of threat is very sought to be topics for researchers[2] .Different works have been delivered in this domain which has been mentioned and discussed in the related work segment. The one of the dimension of Semantic Web mining is implementing Google search algorithms which are very random and frequents in approach.[3] The other dimension is to implement artificial malware detection, to detect plagiarism in the content using content mining. In order to perform all the mentioned applications proper pattern analysis should be done in designed dataset that will improve the accuracy of the results using classification algorithms.[4]

II. RELATED WORK

Keerthana. B, Sivashankari.K and Shaistha Tabasum.S (2018) developed a system, Detecting Malwares and Search Rank Fraud in Google Search using Rabin Karp Algorithm. Here, they used Rabin Karp algorithm to detect prototype in strings and at the same time, Java offers for the most part of powerful API's like IO functions such as reading, writing as well as searching the file, counting the keywords, matching and so on. To find plagiarism by compare strings in document in the midst of supplementary strings in document via use Rabin Karp algorithm. It is used to detect the contented feature in Google as well as to facilitate Google's search algorithm meant for improved accuracy [1].

Iker Burguera and Urko Zurutuza and Simin Nadjm-Tehrani (2011) discussed a theory about Crowdroid: Behavior-Based Malware Detection System for Android. In this paper, they established by investigate the statistics composed in the middle server by means of two types of data sets; for test basis they produced as of those artificial malware as well as those as of real malware originate in the wild. These techniques illustrate the probable used for keep away from the scattering to identify the malware to a better community. They proposed this system to attain as well as evaluate smart phone application action [2].

Michael Grace, Yajin Zhou, Qiang Zhang, Shihong Zou and Xuxian Jiang (2012) developed a system based on RiskRanker: Scalable and Accurate Zero-day Android Malware Detection. Our method is aggravated to review probable protection threat pretense via these untrusted apps by devoid of relying on malware samples as well as their name. To analyze the scalable, the authors proposed a programmed system known as RiskRanker explicitly, whether a specific app shows hazardous activities. The above mentioned outputs exhibit the efficiency as well as scalability of RiskRanker towards regulates Android markets of the entire stripes [3].

Lemon Akoglu, Rishi Chandy and Christos Faloustos (2013) discussed a framework about Opinion Fraud Detection in Online Reviews by Network Effects. A framework was proposed by the authors to detect fraudsters as well as fake reviews in online review datasets called FRAUDEAGLE. The proposed system has many advantages which are followed by,

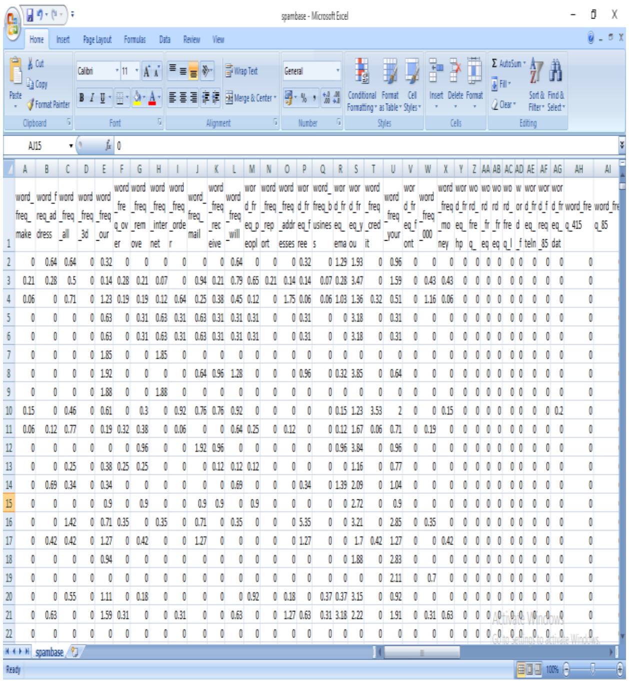
(i) it exploits the network effect among reviewers as well as products, (ii) for large datasets, it is scalable and at the same time among the network size, the run time of our proposed system increases linearly, (iii) the unsupervised fashion requires no labeled data which was operated by our proposed system. Furthermore, the performance of the system is much better [10].

III. METHODOLOGY

The Weka tool has been used for the implementation part over a dataset mentioning all mail details along with spam and important or relevant mail. The concept of web mining is justified basically but the work has not been implemented over web medium. First we need all the attributes information in order to avoid any malicious content which will increase complexity. The methods that have been used through the revolutionary classifier called naïve Bayes classifier in order to get extensive classification. Our work is basically based on eliminating spam on the basis of pre built parameter by using naïve Bayes classifier. The classifications performed over different dimension like cross validation, attribute classifications, visualization of classification

IV. EXPERIMENTS RESULTS

To select a file we use preprocess tab



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	word_freq_make	word_freq_address	word_freq_all	word_freq_3d	word_freq_our	word_freq_over	word_freq_remove	word_freq_internet	word_freq_order	word_freq_mail	word_freq_receive	word_freq_will	word_freq_people	word_freq_report	word_freq_addresses	word_freq_free	word_freq_business	word_freq_email	word_freq_you	word_freq_credit	word_freq_your	word_freq_font	word_freq_000	word_freq_money	word_freq_hp	word_freq_	word_freq	word_freq	word_freq_l	word_f	word_freq_teln	word_freq_85	word_freq_dat	word_freq_415	word_freq_85
2	0	0.64	0.64	0	0.32	0	0	0	0	0	0	0.64	0	0	0	0.32	0	1.29	1.93	0	0.96	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0.21	0.28	0.5	0	0.14	0.28	0.21	0.07	0	0.94	0.21	0.79	0.65	0.21	0.14	0.14	0.07	0.28	3.47	0	1.59	0	0.43	0.43	0	0	0	0	0	0	0	0	0	0	
4	0.06	0	0.71	0	1.23	0.19	0.19	0.12	0.64	0.25	0.38	0.45	0.12	0	1.75	0.06	0.06	1.03	1.36	0.32	0.51	0	1.16	0.06	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0.63	0	0.31	0.63	0.31	0.63	0.31	0.31	0.31	0	0	0.31	0	0	3.18	0	0.31	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0.63	0	0.31	0.63	0.31	0.63	0.31	0.31	0.31	0	0	0.31	0	0	3.18	0	0.31	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	1.85	0	0	1.85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	1.92	0	0	0	0	0.64	0.96	1.28	0	0	0	0.96	0	0.32	3.85	0	0.64	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	1.88	0	0	1.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0.15	0	0.46	0	0.61	0	0.3	0	0.92	0.76	0.76	0.92	0	0	0	0	0	0.15	1.23	3.53	2	0	0	0.15	0	0	0	0	0	0	0	0	0.2	0	
11	0.06	0.12	0.77	0	0.19	0.32	0.38	0	0.06	0	0	0.64	0.25	0	0.12	0	0	0.12	1.67	0.06	0.71	0	0.19	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0.96	0	0	1.92	0.96	0	0	0	0	0	0	0.96	3.84	0	0.96	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0.25	0	0.38	0.25	0.25	0	0	0	0.12	0.12	0.12	0	0	0	0	0	1.16	0	0.77	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0.69	0.34	0	0.34	0	0	0	0	0	0	0.69	0	0	0	0.34	0	1.39	2.09	0	1.04	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0.9	0	0.9	0	0	0.9	0.9	0	0.9	0	0	0	0	0	2.72	0	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	1.42	0	0.71	0.35	0	0.35	0	0.71	0	0.35	0	0	0	5.35	0	0	3.21	0	2.85	0	0.35	0	0	0	0	0	0	0	0	0	0	0	
17	0	0.42	0.42	0	1.27	0	0.42	0	0	1.27	0	0	0	0	0	1.27	0	0	1.7	0.42	1.27	0	0	0.42	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0.94	0	0	0	0	0	0	0	0	0	0	0	0	0	1.88	0	2.83	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.11	0	0.7	0	0	0	0	0	0	0	0	0	0	0	
20	0	0	0.55	0	1.11	0	0.18	0	0	0	0	0	0.92	0	0.18	0	0.37	0.37	3.15	0	0.92	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	0	0.63	0	0	1.59	0.31	0	0	0.31	0	0	0.63	0	0	1.27	0.63	0.31	3.18	2.22	0	1.91	0	0.31	0.63	0	0	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 1: dataset

The explanation of dataset is as follows:

1. (a) Determine whether a given email is spam or not.
   (b) ~7 misclassification error.
   False positives (marking good mail as spam) are very undesirable.
   If we insist on zero false positives in the training/testing set,
   20-25 of the spam passed through the filter.

2. Relevant Information:

The "spam" concept is diverse: advertisements for products/websites, make money fast schemes, chain letters, pornography... Our collection of spam e-mails came from our postmaster and individuals who had filed spam. Our collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.

3. Number of Instances: 4601 (1813 Spam = 39.4)

4. Attribute Information:

The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail. Most of the attributes indicate whether a particular word or character was frequently occuring in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters. For the statistical measures of each attribute, see the end of this file. Here are the definitions of the attributes:

(i) 48 continuous real [0,100] attributes of type word_freq_WORD = percentage of words in the e-mail that match WORD,
i.e. 100 * (number of times the WORD appears in the e-mail) / total number of words in e-mail.
A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.
(ii) 6 continuous real [0,100] attributes of type char_freq_CHAR = percentage of characters in the e-mail that match CHAR,
i.e. 100 * (number of CHAR occurences) / total characters in e-mail
(iii) 1 continuous real [1,...] attribute of type capital_run_length_average = average length of uninterrupted sequences of capital letters
(iv) 1 continuous integer [1,...] attribute of type capital_run_length_longest = length of longest uninterrupted sequence of capital letters
(v) 1 continuous integer [1,...] attribute of type capital_run_length_total = sum of length of uninterrupted sequences of capital letters = total number of capital letters in the e-mail
(vi)1 nominal {0,1} class attribute of type spam = denotes whether the e-mail was considered spam (1) or not (0),
i.e. unsolicited commercial e-mail.

5. Missing Attribute Values: None
6. Class Distribution:

Spam 1813 (39.4)

Non-Spam 2788 (60.6)

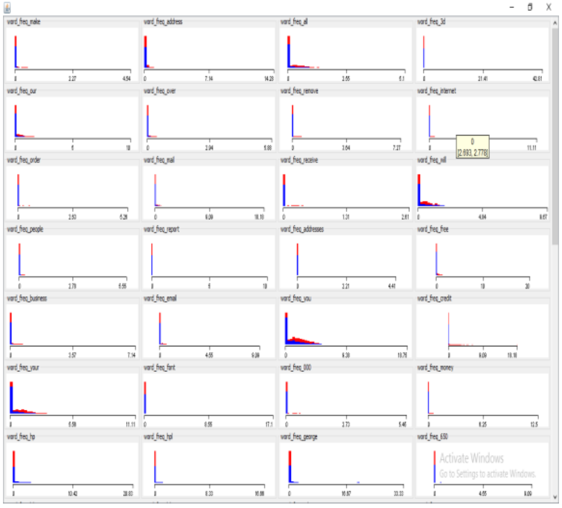


Figure 2.1: visualization of attributes in weka

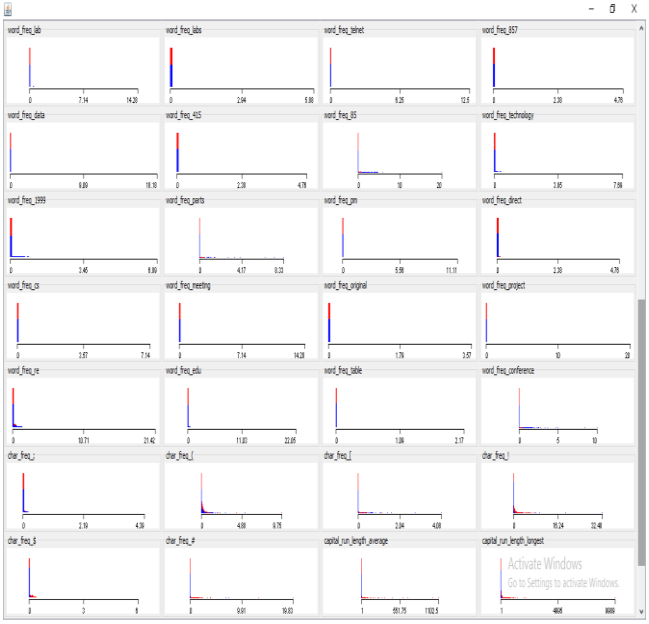


Figure 2.2 visualization of attributes in weka

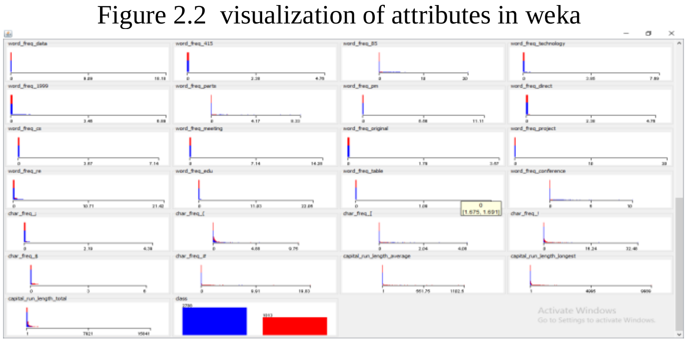


Figure 2.3: visualization of attributes in weka

These all visualizations show the minimum, maximum, mean and std. Deviation value of the each attribute in graphical form. We can also get the information about the data type of the attribute in this section. Example is given below in figure 3.

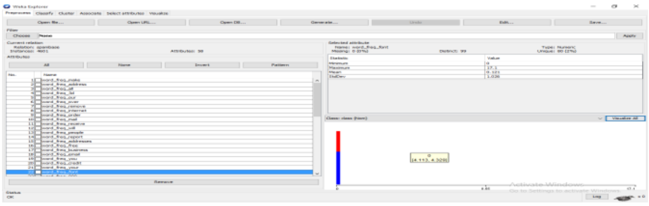
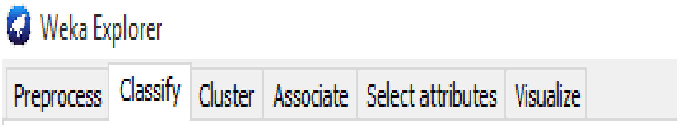
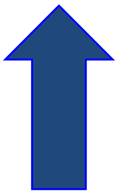


Figure 3: Statistical information of an attribute

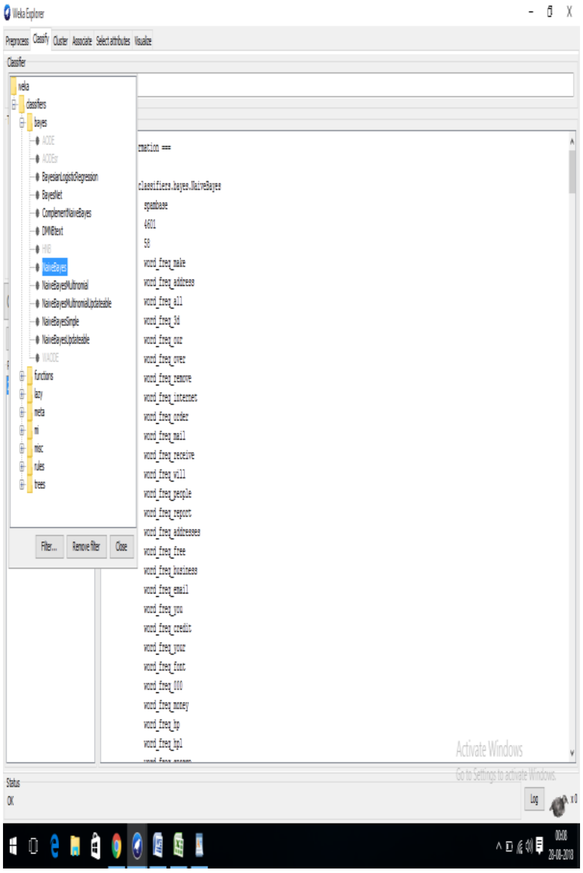
Naive Bayes Algorithm implementation

1. Select Classify Tab





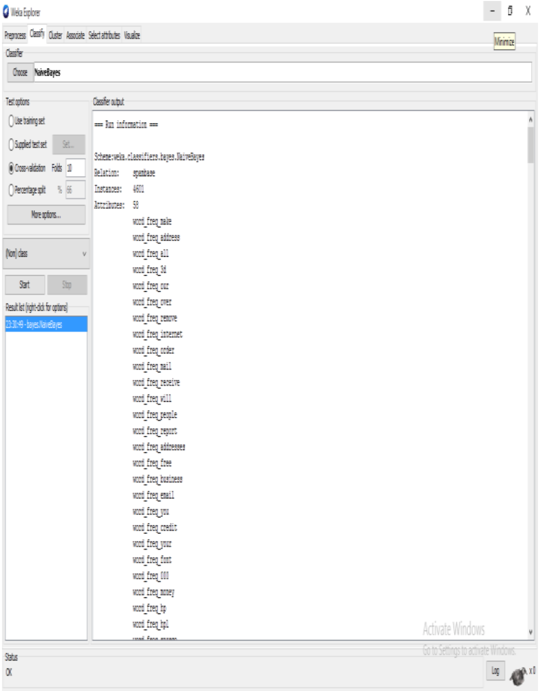
2. Choose Naive Byes from the classifiers list. And then Press Start button.



```
rmation ==

classifiers.bayes.NaiveBayes
  spambase
  4601
  58
  word_freq_make
  word_freq_address
  word_freq_all
  word_freq_3d
  word_freq_our
  word_freq_over
  word_freq_remove
  word_freq_internet
  word_freq_order
  word_freq_mail
  word_freq_receive
  word_freq_will
  word_freq_people
  word_freq_report
  word_freq_addresses
  word_freq_free
  word_freq_business
  word_freq_email
  word_freq_you
  word_freq_credit
  word_freq_your
  word_freq_font
  word_freq_000
  word_freq_money
  word_freq_hp
  word_freq_hpl
```

Figure 4: Classifier Selection

3. The classifier output will be shown as below:



Choose: NaiveBayes

Cross-validation Folds: 10

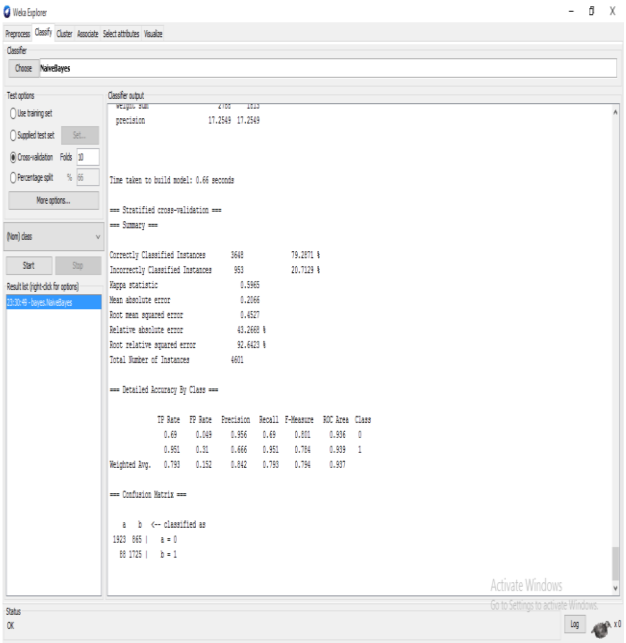
```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     spambase
Instances:    4601
Attributes:   58
              word_freq_make
              word_freq_address
              word_freq_all
              word_freq_3d
              word_freq_our
              word_freq_over
              word_freq_remove
              word_freq_internet
              word_freq_order
              word_freq_mail
              word_freq_receive
              word_freq_will
              word_freq_people
              word_freq_report
              word_freq_addresses
              word_freq_free
              word_freq_business
              word_freq_email
              word_freq_you
              word_freq_credit
              word_freq_your
              word_freq_font
              word_freq_000
              word_freq_money
              word_freq_hp
              word_freq_hpl
```

Figure 5.1 :Classifier Output

4. The below table shows the classified information of each attribute according to the class attribute.

5. Here is the stratified cross-validation summary. Which show the correctly, incorrectly instances of the dataset. It also shows the summary of various classification errors like Mean absolute error, Root Mean Squared errors, Relative absolute errors, Root relative squared error etc. And then we can see the detailed accuracy of the dataset value according to the classes (Decision Attributes) in the form of TP, FP rate, Precision, Recall, F-measure, ROC Area etc.

And the last it shows the Confusion Matrix for the final result. This is 2X2 matrix generated for each class. It is generated from the TP and FP rates.



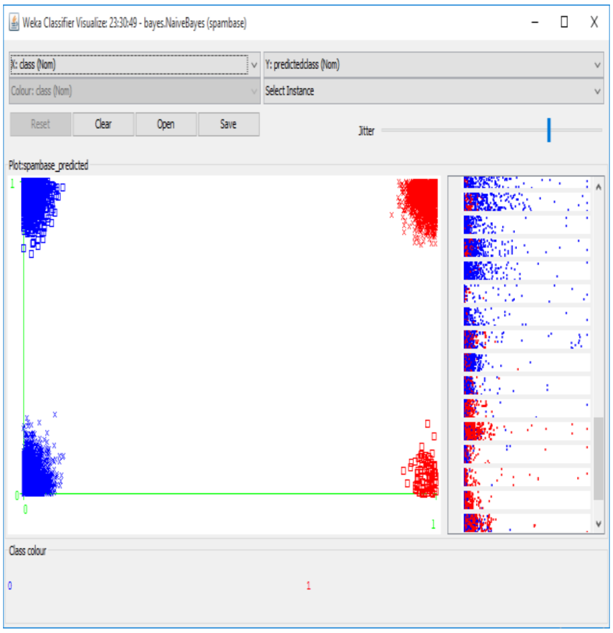
Choose: NaiveBayes

Cross-validation Folds: 10

```
 precision              17.2549  17.2549



Time taken to build model: 0.66 seconds

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

Correctly Classified Instances       3648               79.2871 %
Incorrectly Classified Instances      953               20.7129 %
Kappa statistic                          0.5965
Mean absolute error                      0.2066
Root mean squared error                  0.4527
Relative absolute error                 43.2668 %
Root relative squared error             92.6423 %
Total Number of Instances             4601

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.69      0.049     0.956      0.69      0.801      0.936    0
                 0.951     0.31      0.666      0.951     0.784      0.939    1
Weighted Avg.    0.793     0.152     0.842      0.793     0.794      0.937

=== Confusion Matrix ===

    a    b   <-- classified as
 1923  865 |    a = 0
   88 1725 |    b = 1
```

Figure 6: Stratified cross validation summary

6. We can also visualize these errors in the form of graphical visualization shown as below:



Weka Classifier Visualize: 23:30:49 - bayes.NaiveBayes (spambase)

X: class (Nom)

Y: predictedclass (Nom)

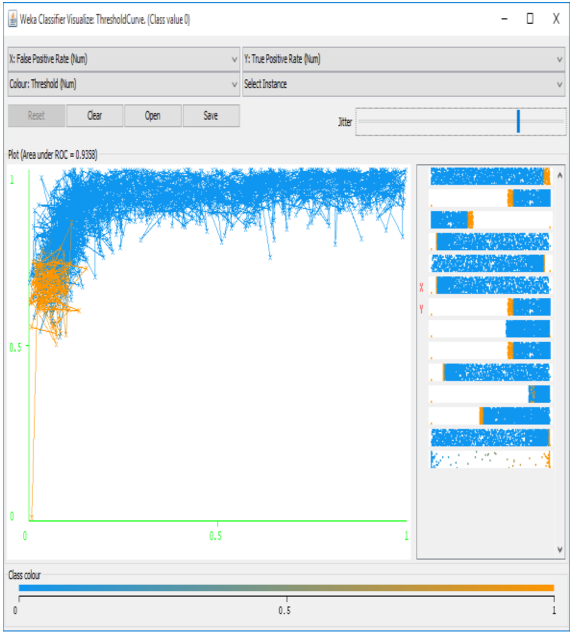
Colour: class (Nom)

Select Instance

Plot:spambase_predicted

Figure 7: Visualization of Classifier errors

7. We can see the Threshold curve for both classes in graphical visualization shown as below:



Weka Classifier Visualize: ThresholdCurve. (Class value 0)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.9358)

Figure 8: Visualization of Threshold curve for each class

8. Final Predicted result:
(word_freq_hp <= 0.02) and (capital_run_length_longest >= 10) and (word_freq_edu <= 0) and (char_freq_$ >= 0.015) => class=1 (205.0/19.0)

In above rule, we can say that the email which has frequency of word hp is less than <0.02 and capital run length longes is greater than 10 and frequency of word edu is less than 0 and character frequency of $ is greater than 0.015 then these kind of emails are to be considered in the spam email category.

V. *CONCLUSION*

Web mining work is very much interesting and getting very applicable in real time system. Our work is based on semantic web mining but at the same time, security measures and threats have been discussed with implementation results. Different case studies and implemented work has been mentioned and discussed regarding malware detections, plagiarism detection and content mining using World Wide Web. Our work is not limited around Semantic web mining only but providing securities measures along with their pattern analysis using classifiers. The work is based on a dataset which has been designed on the basis of spam on the mail. The repeated pattern of that has been monitored and detected with accuracy in results. The future scope can be more based on web approach like working over online data. Several case studies has already been mentioned in past work .Our work presents the collaborative approach for semantic web mining as well as outliers and malicious data detection like spam and irrelevant mails.

References

1. E. Siegel, “Fake reviews in Google Play and Apple App Store,” Appentive, Seattle, WA, USA, 2014.

2. I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani , “Crowdroid: Behavior-based Malware detection system for Android,” in Proc. ACM SPSM, 2011, pp. 15–26.

3. M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, “RiskRanker: Scalable and accurate zero-day Android malware detection,” in Proc. ACM MobiSys, 2012, pp. 281–294.

4. B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, “Android Permissions: A Perspective Combining Risks and Benefits,” in Proc. 17th ACM Symp. Access Control Models Technol., 2012, pp. 13–22.

5. S. Yerima, S. Sezer, and I. Muttik, “Android Malware detection using parallel machine learning classifiers,” in Proc. NGMAST, Sep. 2014, pp. 37–42.

6. Y. Zhou and X. Jiang, “Dissecting Android malware: Characterization and evolution,” in Proc. IEEE Symp. Secur. Privacy, 2012, pp. 95–109.

7. J. Sahs and L. Khan, “A machine learning approach to Android malware detection,” in Proc. Eur. Intell. Secur. Inf. Conf., 2012, pp. 141–147.

8. B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas, and G. Alvarez, “Puma: Permission usage to detect malware in android,” in Proc. Int. Joint Conf. CISIS12-ICEUTE’ 12-SOCO’ Special Sessions, 2013, pp. 289–298.

9. J. Ye and L. Akoglu, “Discovering opinion spammer groups by network footprints,” in Machine Learning and Knowledge Discovery in Databases. Berlin, Germany: Springer, 2015, pp. 267–282.

10. L. Akoglu, R. Chandy, and C. Faloutsos, “Opinion Fraud Detection in Online Reviews by Network Effects,” in Proc. 7th Int. AAAI Conf. Weblogs Soc. Media, 2013, pp. 2–11.