JOIN (Jurnal Online Informatika)
Volume 4 No. 1 | Juni 2019 : 53-56
DOI: 10.15575/join.v4i1.298

# An Analysis of Spam Email Detection Performance Assessment Using Machine Learning

**Budi Santoso**
Engineering Faculty, Universitas Dr. Soetomo, Surabaya, Indonesia
budi.santoso@unitomo.ac.id

**Abstract- Spam email is very annoying for email account users to get relevant information. Detection of email spam has actually been applied to email services for the public with various methods. But for the use of a limited number of company's e-mail accounts, not all e-mail servers provide spam e-mail detection features. The server administrator must add a separate or modular spam detection feature so that e-mail accounts can be protected from spam e-mail. This study aims to get the best method in the process of detecting spam emails. Some machine learning methods such as Logistic Regression, Decision Tree, and Random Forest are applied and compared results to get the most efficient method of detecting spam e-mail. Efficiency measurements are obtained from the speed of training and testing processes, as well as the accuracy in detecting spam emails. The results obtained in this study indicate that the Random Forest method has the best performance with a test data speed of 0.19 seconds and an accuracy of 98%. This result can be used as a reference for the development of spam detection using other methods.**

***Keywords*- spam detection, e-mail, machine learning, performance**

## I. INTRODUCTION

Certain purposes of email abuse that send irrelevant information (Spam) is often faced by all email account owner even though some of the email users can identify real email or spam. Nowadays the activity of spamming is continuing with another more different variety of its purposes, however, there are a bunch of annoying spam fulfilling user's email inbox and at least it will be such a waste of time if user should identify one-by-one manually by reading those spams [1].

Some of the email services has added spam detection feature automatically based on prior email sent history as the basis of identification process. Meanwhile, there are still many of email server particularly organized personally by a company that has not add spam detection feature to their server. That is caused by not all email server product in a default status to provide spam detection feature as one of their software installation modules. So that the process of spam identification should be made and inserted as partial software or additional module [2].

There are some ways of spam identification that have researched by several researchers such as the implementation of a k-NN method by Pratiwi et al. [1], As well as application of Fuzzy logic by [3]. This method looks forward to its cluster according to membership function of fuzzy in each email.

Machine learning is one of the Artificial Intelligent that could recognize a pattern based on the process of learning/training from several determined input data[4]–[6]. There are some methods in machine learning that used to recognize spam email. In this research, some of those mentioned methods had applied and tested to a bunch of spam email dataset. The machine learning method testing in this research tends to obtain a perfect method so that it can be applied as a spam email detection feature efficiently.

## II. METHOD

This research is using three methods in *machine learning* that conducted partially to detect spam, there are Logistic Regression, Decision Tree and Random Forest methods.

### A. Logistic Regression

Logistic Regression is stated as an approach to create a prediction model such as linear regression or *Ordinary Least Squares* (OLS) [7].

Logistic regression formularization is given to the equation of (1) and (2).

$$\ln\left(\frac{\tilde{p}}{1-\tilde{p}}\right) = \beta_0 + \beta_1 X \text{ ................ (1)}$$

Notes:

- $\beta_0 + \beta_1 X$ is an equation from OLS
- $\tilde{p}$ is a logistic probability

Logistic probability is obtained:

$$\tilde{p} = \frac{e^{(\beta_0+\beta_1 X)}}{1+e^{(\beta_0+\beta_1 X)}} \text{ .................. (2)}$$

Notes:

- $\tilde{p}$ is logistic probability
- $\beta_0$ is constant
- $\beta_1$ is regression coefficient
- X is free variable

### B. Decision Tree

A decision tree is a method that used not only in data mining but also in machine learning to classify the decision tree.

Received: April 29, 2019; Revised: June 08, 2019; Received: August 31, 2019; Published: August 31,2019

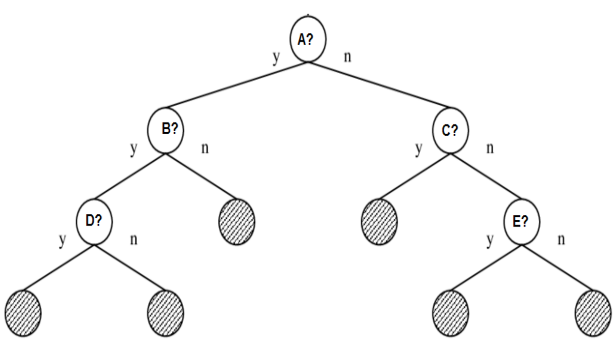


Figure 1. Decision Tree Illustration

The classified input data has several characteristics as follows:

1. The data or example stated by attribute's pair and its value
2. Label or output data usually valued discrete
3. Data has a missing value

In order to comprehend the decision tree, researcher made a set of rules such as if-then where one rule generalized into all over knots. Figure 1 displays decision tree illustration where a rule with its similar structure but in different attribute[8]. Several popular method development of decision tree is ID3 and C.45.

One main excellency in the use of decision tree method is the process of eliminating unnecessary mathematical calculations because the sample test is only based on certain criteria or class.

Meanwhile, the decision tree still has weakness if there are many classes or criteria. This is can cause overlap and increasing decision taking the time that needs a lot of computerizing memory consumption at the same time. Besides the quality of decision taking from this method is highly depends on tree design that has made [9].

### C. Random Forest

Random forest method is a next-level version of the CART method that applied bootstrap aggregating (bagging) method and random feature selection[10], [11]. In this method, the forest is formed from many trees then analyze to a bunch of trees to obtain classification input data result [12]–[14].

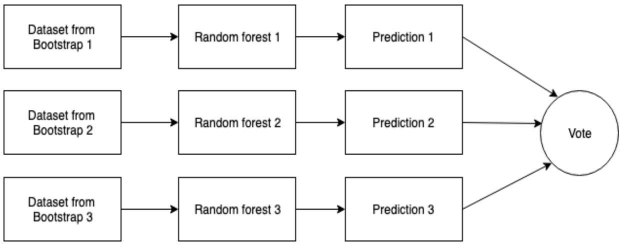


Figure 2. Random forest illustration

The final result of classification assessment input is determined by the result majority in each prediction phase such as illustrated in figure 2.

In some cases, the implementation of Random Forest has more beneficial particularly in producing smaller-rate of error. Besides that, it is also can handle some of missing data which is the circumstance consists some of the training data that has empty value on its feature.

Meanwhile because of Random Forest is made from the decision tree in a big scale so that it has inevitably major weakness particularly in the time of computerize process where if it only uses single processor. On the other hand, this weakness can solve by using *parallel processing* to a *multicore computer*.

### *D. Dataset Email Spam*

Email spam data that used as the data training and tested data in this research are mainly using dataset email spam *Code Project Machine Learning and AI Challenge*. This Dataset is given as the resource to contest participants to detect email spam efficiently.

Data then labeled as spam or not spam in order to support the process of identification result validation with using machine learning method determined result.

### *E. Confusion Matrix*

In this research, the system testing is in accuracy evaluation in the process of spam classification towards the dataset by using a confusion matrix[15]. An illustration regarding the confusion matrix then shown in Table 1.

Table 1. Confusion Matrix

| | | Prediction Result Label | |
|---|---|---|---|
| | | Negative | Positive |
| Authentic/True Label | Negative | True Negative (TN) | False Negative (FN) |
| | Positive | False Positive (FP) | True Positive(TP) |

Notes:

- True Negative (TN) is a number of true negative data categorized as a negative label
- False Negative (FN) is a number of negative data that categorized as a positive label
- False Positive (FP) is a number of positive data that categorized as a negative label
- True Positive (TP) is a number of true positive data that categorized as a positive label

From the table of confusion matrix above, then it conducts a calculation to obtain accuracy level, recall, *precision,* and *F-measure*.

$$\text{Accuracy} = \frac{(\text{TN+TP})}{(\text{TN+FP+FN+TP})} \ldots\ldots\ldots\ldots\ldots (3)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{FPTP})} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{FN}+\text{TP})} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5)$$

$$\text{FMeasure} = \frac{2*\text{TP}}{(2*\text{TP}+\text{FP}+\text{FN})} \ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

*F. System Design*

The spam email testing system using machine learning could be seen in figure 3. The first step is preparing data that covers parsing data and split. This process aimed to separate half data as the data training email spam, data training email non-spam (ham) and the data test.

.

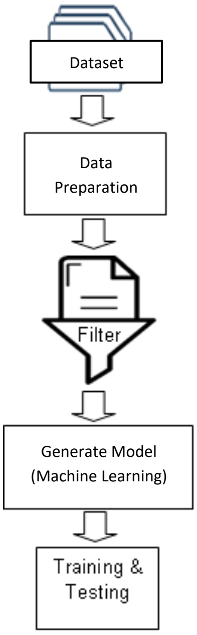


Figure 3. System chart diagram

The next phase is data filtering process to remove unnecessary words or meaningless phrases. Then the researcher created a model that represents each determined method. The final step is the process of training and modeling testing that have made to acquired accuracy value from spam email identification.

Besides accuracy value, another perimeter as the comparison is the speed of training process that using existing dataset.

## III. RESULT AND DISCUSSION

In the first step of data preparation it has obtained that in the dataset which consists of 2000 email that divided into 1000 identified email as spam, as well as another 1000 email identified as relevant email (non-spam) as could be seen in figure 4.

.

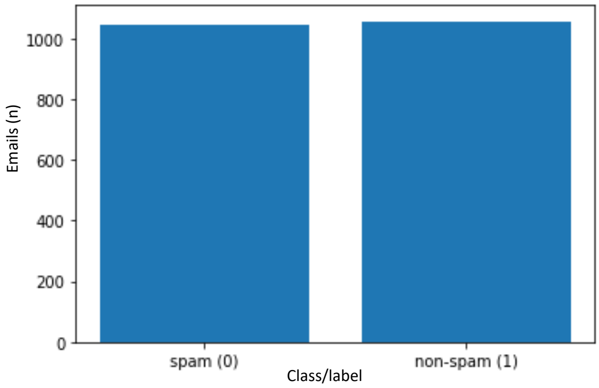


Figure 4. Distribution label dataset email spam

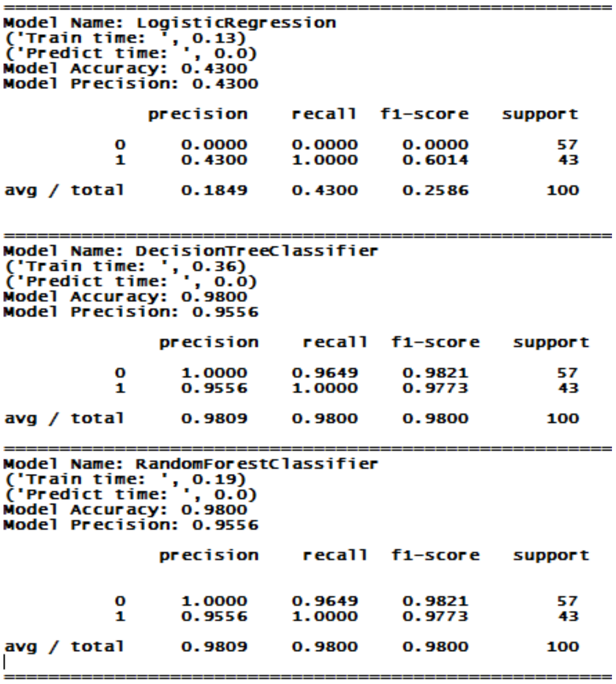
Both those data categories then split as the random data training to get the test data. In the second phase that is filtering process each email from meaningless words or irrelevant to the whole sentence in the email's content.

From this process then obtained results as in the Table 1. Those previous method then each created modeling based on existing data training from the prior process.

The final step as the distribution of data test input to each *machine learning* model with the Logistic Regression method, Decision Tree, and Random Forest. The output process of accuracy calculation from each method are using Python and PC with specification Intel Core2Duo 2.1 GHz as shown in figure 5.

Table 1. The Comparison of Before and After Data Filtering

| Label | Before *Filtering* | After *Filtering* |
|---|---|---|
| 1 | Spam,<p>But could then once pomp to nor that g... | But could then once pomp to nor that g... |
| 1 | Spam,<p>His honeyed and land vile are so and n... | His honeyed and land vile are so and n... |
| 1 | Spam,<p>Tear womans his was by had tis her ere... | Tear womans his was by had tis her ere... |
| 1 | Spam,<p>The that and land. Cell shun blazon pa... | The that and land. Cell shun blazon pa... |
| 1 | Spam,<p>Sing aught through partings things was... | Sing aught through partings things was... |
| 1 | Spam,<p>He den blazon would did prose to he de... | He den blazon would did prose to he de... |



```
=====================================================
Model Name: LogisticRegression
('Train time: ', 0.13)
('Predict time: ', 0.0)
Model Accuracy: 0.4300
Model Precision: 0.4300

             precision    recall  f1-score   support

          0     0.0000    0.0000    0.0000        57
          1     0.4300    1.0000    0.6014        43

avg / total     0.1849    0.4300    0.2586       100

=====================================================
Model Name: DecisionTreeClassifier
('Train time: ', 0.36)
('Predict time: ', 0.0)
Model Accuracy: 0.9800
Model Precision: 0.9556

              precision    recall  f1-score   support

          0     1.0000    0.9649    0.9821        57
          1     0.9556    1.0000    0.9773        43

avg / total     0.9809    0.9800    0.9800       100
=====================================================
Model Name: RandomForestClassifier
('Train time: ', 0.19)
('Predict time: ', 0.0)
Model Accuracy: 0.9800
Model Precision: 0.9556

              precision    recall  f1-score   support


          0     1.0000    0.9649    0.9821        57
          1     0.9556    1.0000    0.9773        43

avg / total     0.9809    0.9800    0.9800       100
|
=====================================================
```

Figure 5. Modeling Result and the calculation from each method

Generally, the comparison results of performance assessment from each method to detect spam portrayed in table 2.

Table 2. The comparison result of performance assessment using a machine learning method

| Method | Training Time(dt) | Accuracy | F1 score |
|---|---|---|---|
| Logistic Regression | 0.13 | 0.43 | 0.2586 |
| Decision Tree | 0.36 | 0.98 | 0.98 |
| Random Forest | 0.19 | 0.98 | 0.98 |

The F1 score is obtained from weight average *precision* and *recall*. Meanwhile, recall is obtained from the positive observation ratio that predicted for all observation from its category (from all messages that actually spam, and how much that can be identified properly) as well as the formula (3),(4),(5), and (6).

This research can compare 3 methods of machine learning perfectly and also obtained an efficient method to identify spam email that is the Random Forest method.

## IV. CONCLUSION

After organized some of the processes and tests as well as the research design, in a sum up the Machine Learning method and Random Forest is able to give satisfying performance assessment with its speediness of training process up to 0,19 seconds and accuracy around 98%.

In future research, the researcher expected that another more complex method such as the ensemble method and the dataset process are more applied so that can contribute a better performance.

## V. REFERENCES

[1] S. N. D. Pratiwi and B. S. S. Ulama, "Klasifikasi Email Spam dengan Menggunakan Metode Support Vector Machine dan k-Nearest Neighbor," *J. SAINS DAN SENI ITS*, vol. 5, no. 2, 2016.

[2] A. Saputra and M. Syafrizal, "Perancangan dan Implementasi Mail Server pada CV. Sanjaya Anugerah Sejahtera (Isp Jogjaringan) Berbasis Open Source," *J. DASI*, vol. 13, no. 2, 2012.

[3] F. Rozi and R. Kartadie, "Deteksi E-Mail dan Spam Menggunakan Fuzzy Association Rule Mining," *J. Ilm. Penelit. dan Pembelajaran Inform.*, vol. 02, no. 02, 2017.

[4] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[5] N. M. Samsudin, C. F. binti Mohd Foozy, N. Alias, P. Shamala, N. F. Othman, and W. I. S. Wan Din, "Youtube spam detection framework using naïve bayes and logistic regression," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 3, p. 1508, Jun. 2019.

[6] N. Alias, C. F. M. Foozy, S. N. Ramli, and N. Zainuddin, "Video spam comment features selection using machine learning techniques," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 2, pp. 1046–1053, Aug. 2019.

[7] A. T. Basuki, *Bahan Ajar Ekonometrika*. Yogyakarta: Universitas Muhammadiyah Yogyakarta, 2017.

[8] K. Hastuti and E. Y. Hidayat, "Analisis Algoritma Decision Tree untuk Prediksi Mahasiswa Non Aktif," 2013.

[9] A. Saputra, *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. 2016.

[10] X. Luo, "A New Text Classifier Based on Random Forests," vol. 107, no. Meita 2016, pp. 290–293, 2017.

[11] T. T. A. Putri, H. W. S, I. Y. Sitepu, M. Sihombing, and Silvi, "Analysis and Detection of Hoax Contents in Indonesian News Based on Machine Learning," *JIPN (Journal Informatics Pelita Nusantara)*, vol. 4, no. 1, pp. 19–26, 2019.

[12] S. S. Pangastuti, "Perbandingan Metode Ensemble Random Forest dengan Smote-Boosting dan Smote-Bagging pada Klasifikasi Data Mining untuk Kelas Imbalance," Institut Teknologi Sepuluh Nopember, Surabaya, 2018.

[13] H. W. Nugroho, T. B. Adji, and N. A. Setiawan, "Random Forest Weighting based Feature Selection for C4.5 Algorithm on Wart Treatment Selection Method," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 5, p. 1858, Oct. 2018.

[14] S. Samsuddin, Z. Ali Shah, R. R. Saedudin, S. Kasim, and C. Sen Seah, "Analysis of Attribute Selection and Classification Algorithm Applied to Hepatitis Patients," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, p. 967, May 2019.

[15] A. R. Chrismanto and Y. Lukito, "Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes," *J. Ultim.*, vol. IX, no. 1, 2017.