Logistic Model Tree Induction Machine Learning Technique for Email Spam Filtering

Emmanuel Gbenga Dada, Ph.D.* **and Joseph Stephen Bassi, Ph.D.**

Department of Computer Engineering, University of Maiduguri, Maiduguri, Borno State, Nigeria.

Email: gbengadada@unimaid.edu.ng*

ABSTRACT

The susceptible characteristics of email spams allow them to undergo changes that can make them to easily evade spam filters. This necessitates the need to develop more effective spam filters. Machine learning approaches have proved to be an efficient method for solving the problem of several spam emails wreaking havoc on email users. The conventional techniques of spam filtering like black lists and white lists (using domains, IP addresses, mailing addresses, etc.) have not been able to effectively curb the hazards posed by spam emails. In this paper, we applied the Logistic Model Tree machine learning algorithm for efficient classification of email spam messages.

The aim of this study is to develop an email spam filter with superior prediction accuracy and fewer number of features. From the Enron public dataset consisting of 5,180 emails of both ham, spam, and normal emails, some features were extracted and used by the Logistic Model Tree Induction algorithm. Our technique has a classification accuracy of 99.305%, very low false positive rate (0.05), and very high true positive rate of 0.995. All experiments are conducted on WEKA data mining and machine learning simulation environment.

(Keywords: machine learning, spam filtering, Logistic Model Tree Induction, neural networks, decision trees, naïve Bayes)

INTRODUCTION

Recently, emails have become one of the most important media for communications. Unsolicited commercial bulk emails, popularly known as spam, have constituted a big problem on the internet. Spam email have continued to pose serious trouble for email users. The emails are categorized as either spam or ham. Unsolicited emails are called as spam and genuine mails are called as ham. The spammer sending the fraudulent emails, gathers email addresses through different sites, and may include viruses and malicious codes. Spam hinders internet users from maximizing storage capacity and network bandwidth. The presence of large volume of spam mails in computer networks is detrimental to the effective usage of email server memory, bandwidth, CPU processing speed and user time (Fonseca *et al.,* 2016). Reports showed that spam mails are accountable for more than 77% of the email traffic globally (Kaspersky, 2017). Spam emails are very annoying and inimical to users who have fallen victim of phony internet mails and other fraudulent practices of sending emails with the purpose of luring unsuspecting persons to release confidential information such as user names and passwords, Bank Verification Numbers (BVN), and credit card numbers.

Spam filters are deployed by many Internet Service Providers (ISPs) at every layer of the network, in front of email server or at mail relay where there is the presence of firewall. The firewall is a network security system that monitors and manages the incoming and outgoing network traffic based on predetermined security rules. The email server serves as an incorporated anti-spam and anti-virus solution providing a comprehensive safety measure for email at the network perimeter (Savita and Santoshkumar, 2014). Filters can be implemented in clients, where they can be mounted as add-ons, or in computers to serve as intermediary between some endpoint devices (Irwin and Friedman, 2008). Filters block unsolicited or suspicious emails that are a threat to the security of network from getting to the network. At the Local Delivery Agent (LDA) which is a computer software module that delivers email messages to the receiver's inbox, another spam filter can be positioned for the safety of all of clients that

subscribed for the service. Also, at the email level, the user can have a customized spam filter that will block spam emails in accordance with some set conditions (Christina *et al.,* 2010).

To effectively handle the threat posed by email spams, leading email providers such as Gmail, Yahoo mail and Outlook have employed the combination of different machine learning (ML) techniques such as Neural Networks in its spam filters. These ML techniques have the capacity to learn and identify spam mails and phishing messages by analyzing loads of such messages throughout a vast collection of computers. Since machine learning has the capacity to adapt to varying conditions, Gmail and Yahoo mail spam filters do more than just check junk emails using pre-existing rules. They generate new rules themselves based on what they have learned as they continue in their spam filtering operation.

Several research works have been published in literature that proposed various approaches to email spam filtering. Many have been successfully applied to classify emails into either spam or non-spam. These techniques include probabilistic, decision tree, artificial immune system (Bahgat, Rady and Gad, 2016), support vector machine (SVM) (Bouguila and Amayri, 2009), artificial neural networks (ANN) (Cao, Liao and Li, 2004), and case-based techniques (Fdez-Riverola, 2007). It has been demonstrated that it is possible to use these machine learning techniques to filter out spam mails by employing content-based filtering approach that have the ability to identify particular features in email messages (usually keywords frequently used in spam emails). The frequency at which these features occur in emails determine the likelihood that the email will be classified as spam when measured against the threshold value. Email messages that exceed the threshold value are classified as spam (Mason, 2003).

Karthika and Visalakshi (2015) compared the performance of hybridized ACO and SVM with KNN, NB and SVM algorithms on spambase dataset taken from UCI repository. Awad and Foqaha (2016) evaluated the performance of PSO, RBFNN, MLP and ANN using the UCI spambase dataset. Sharma and Suryawanshi (2016) compared the performance of kNN with spearman and kNN with Euclidean using the spambase dataset taken from UCI repository. Awad and ELseuofi (2011) reviewed six state of the art machine learning methods (Bayesian classification, k-NN, ANNs, SVMs, Artificial Immune System and Rough sets) and their applicability to the problem of spam email classification. Alkaht and Al-Khatib (2016) compared the performance of NN, MLP, Perceptron on dataset based on randomly collected emails. Dhanaraj and Palaniswami (2014) evaluated the performance of Firefly, NB, NN and PSO algorithm on CSDMC2010 spam corpus dataset. Palanisamy, Kumaresan and Varalakshmi (2016) compared the performance of NSA, PSO, SVM, NB and DFS-SVM using Ling spam dataset. Zavvar, Rezaei and Garavand (2016) compared the performance of PSO, SOM, kNN and SVM on spambase datasets retrieved from UCI repository. Sosa (2010) evaluated the performance of Sinespam, a spam classification technique using machine learning a corpus of 2200 e-mails from several senders to various receivers gathered by the ISP.

Akshita (2016) applied the Deep Learning technique to content-based spam classification. The author used DL4J deep network on PU1, PU2, PU3, PUA and Enron spam datasets. The main problem with many of these techniques discussed above is the low performance of the filters and there is need to increase the classification accuracy of the filters. Also, many of them are not robust and find it difficult to cope with the evolving nature of spams.

MATERIALS AND METHODS

Most of the email spam filtering techniques use text classification approaches. Therefore, the performance of spam filters are usually below average and find it very difficult to block spam mails from entering the inbox of the users. This study uses rules from the Logistic Model Tree Induction (LMT) algorithm to extract essential features from emails, and group them into either ham, spam or normal.

The Enron spam dataset was used as the benchmark dataset. The Logistic Model Tree Induction machine learning algorithm was simulated using WEKA (Wang, 2005). WEKA have a set of machine learning algorithms that can be used for data preprocessing, classification, regression, clustering and association rules. Machine learning techniques implemented in WEKA are helpful in solving different real-world problems. The toolkit provides a well-defined structure for researchers and

developers to experiment with different machine learning algorithms, to build and evaluate their models. All experiments were conducted on a machine with a AMD A 10-7300 Radeon R6, 10 Compute Cores 4C+6G, 1.90 GHz, 8.00GB of RAM.

Logistic Model Tree Induction (LMT)

The Logistic Model Tree (LMT) algorithm is a supervised training algorithm that combines the basic technique of decision tree learning with the standard Logistic Regression functions at the leaves (Landwehr *et al.*, 2005) and (Sumner *et al.,* 2005). LMT is a kind of decision tree that have logistic regression models at the leaves. It has proved to be a very efficient classifier has proved with a superior degree of accuracy and reliability in different areas of research. The major disadvantage of this technique is the extreme computational complexity suffered during the stimulating phase of the logistic regression models into the tree.

A prediction model is built by arranging the tree down to the leaf and applying the logistic prediction model related to such leaf. The advantage of the logistic model is that it is simple to interpret and transform in comparison to C 4.5 trees. Furthermore, it has been established that trees created by LMT have a lower size when weighted against those created by C 4.5 induction.

The steps used in LMT algorithm are given below:

Start LMT Algorithm

Input: *X*: number of nodes
N: number of features
Y: number of trees to be grown

Growing Initial Tree

Build linear regression model for root node using Log it Boost algorithm

While termination criteria is not true **do**

For whole dataset Log it Boost is run on the dataset for a fixed number of iterations.
After splitting the dataset,
Build Logistic regression models at the child nodes on the corresponding subsets of dataset using Logic Boost algorithm.
The original weights and probability approximations are taken from the parent node.
Continue splitting and model building till a minimum of 15 samples are at the node and a suitable split is discovered.
Use CART algorithm for Tree pruning
Apply a fusion of training error and penalty term for model complexity to make pruning decisions.

EndWhile

Produce output for each created trees
Use a new sample for each created trees starting from the root node
Assign the sample to the class matching the leaf node.
Output the classification result based on pruning decisions

End LMT Algorithm

Performance Evaluation Measures

We evaluated the performance of LMT on Enron datasets containing ham and spam messages which are publicly available to users. The performance measures we used are as follows:

Assuming:
NH = Number of non-spam messages to be classified
NS = Number of spam messages to be classified

Classification Accuracy

$$(Acc) = \frac{|H \rightarrow H| + |S \rightarrow S|}{N_H + N_S} \quad \textbf{(1)}$$

Classification Error

$$(Err) = 1 - Acc = \frac{|H \rightarrow S| + |S \rightarrow H|}{N_H + N_S} \quad \textbf{(2)}$$

False Positive Rate (FPR): Is the ratio of ham or valid e-mails that are classified as spam

$$FPR = \frac{NoOfFalsePostives}{NoOfFalsePositives + NoOfTrueNegatives} \quad \textbf{(3)}$$

False Negative Rate (FNR): Is the ratio of spam messages that were wrongly classified as ham is called false negative rate (FNR).

$$FNR = \frac{NoOfFalseNegatives}{NoOfTruePositives+NoOfFalseNegatives} \quad \textbf{(4)}$$

$$TPR\ or\ (Recall) = \frac{TP}{TP + FN} \quad \textbf{(5)}$$

Dataset Used for Experiment

The Enron spam datasets was used for our experiment (Koprinska, *et al.,* 2007). The Enron spam datasets from the Enron corporation is used in this study. There are 5180 emails as dataset in three folders: norm for normal, ham for non-spam and spam for Spam emails. Enron has 5180 instances, 3672 ham, 8 norm, and 1500 spam emails. The dataset features are as follows:

i. Some specific word or character was recurrent in the emails.

ii. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.

Normalization is a technique employed to homogenize the measure of autonomous variables or attributes of data. It is usually carried out in the data pre-processing phase. Normalization can be done at the level of the input features or at the level of the kernel (Asa and Jason, 2010). In several applications, the obtainable features are continuous s values, where each feature is measured in a different scale and contains a distinct range of likely values. In that type of situation, it is usually helpful to gauge every feature to a general range by homogenizing the data.

The original dataset used in our experiments consists of 5180 text files. The data contained in those files are not normalized. This means that they have to be normalized before it can serve as input to WEKA. It is required that all data be converted to one .arff file before it can be given to WEKA for training. To achieve this, we use the following command in command line interface of WEKA.

"java weka.core.converters.TextDirectoryLoader -dir D:/Enron > D:/Spam_mails.arff"

After the normalization process, the normalized file was given to WEKA for pre-processing.

FEATURE EXTRACTION

Immediately after the pre-processing stage comes the feature extraction. Feature extraction can be defined is the process of selecting a subset of the terms in the training set and exploiting only this subset as features in text classification. This is accomplished by using some set of rules. Feature extraction makes training and applying a classifier more efficient by decreasing the size of the effective vocabulary and usually enhances classification accuracy by removing noise features. Some of the important email features we used for our spam filtering include: Message body and subject, Volume of the message, Occurrence count of words, Number of semantic discrepancies patterns in the message, Recipient age, Sex and country, Recipient replied, Adult content, Bag of words from the message content, Domain name, IP Address, More blank lines in body.

RESULTS AND DISCUSSION

We present the results of our experiments in this section. The Logistic Model Tree algorithm was applied to classify and evaluate the dataset, we used the 10-fold cross validation test which is an approach employed in appraising predictive models that divide the original set into a training sample to train the model, and a test set for its evaluation. First, the training of the datasets was performed with the feature vectors extracted by analyzing each message header, checking of keywords and whitelist/blacklist. The performance of the trained models is evaluated using 10-fold cross validation for its classification accuracy. Classification accuracy is one of the performance metrics for email spam classification. It is measured as the ratio of number of correctly classified instances in the test dataset and the total number of test cases. In spam filtering, false negatives mean that some spam mails were wrongly classified as non-spam and allowed to enter the user's inbox. False positive mean that non-spam emails were mistakenly classified as spam and moved to spam folder or discarded. For most users, erroneously classifying valid emails as spam can be very costly than receiving spam mails in their inbox. The false positive rate is also one of the performance metrics used in evaluating the effectiveness of email spam filter. Depicted in figure 1 below is the screen shot of our output on WEKA simulation environment.

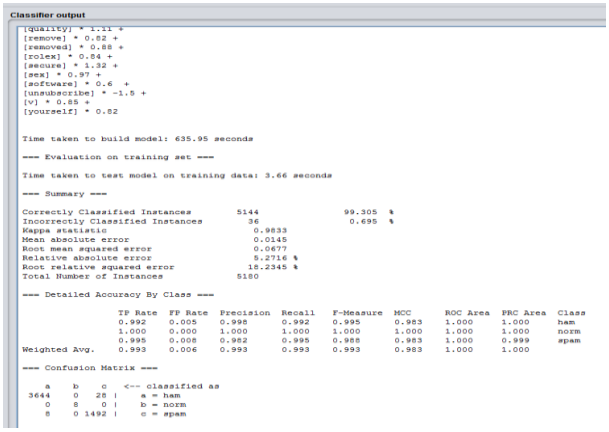

Figure 1: Screen Shot of LMT Classification Output for Enron Spam Emails Datasets.

Table 1: Performance Evaluation of LMT Algorithm.

Evaluation Criteria	Performance
Time taken to create model(s)	635.95
Correctly classified instances	5,144
Incorrectly classified instances	36
Accuracy (%)	99.305

To do a thorough performance evaluation of the LMT algorithm that is under consideration, simulation error is also taken into account in this work. The effectiveness of these algorithms is evaluated using the following terms: Kappa statistic (KS), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), Root Relative Squared Error (RRSE). The KS, MAE and RMSE are in numeric values. RAE and RRSE are in percentage. The results are shown in Table 2.

Table 2. Training and Simulation Error of LMT Algorithms.

Evaluation Criteria	LMT
Kappa Statistics (KS)	0.9833
Mean Absolute Error (MAE)	0.0145
Root Mean Square Error (RMSE)	0.0677
Relative Absolute Error (RAE) %	5.2716
Root Relative Squared Absolute Error (RRAE) %	18.2345

Efficiency

After creating the predictive model, the efficiency of the LMT algorithm was evaluated as shown in Table 3.

Table 3: Performance Evaluation of LMT Algorithms Based on TPR, FTR, Precision, and F-Score.

Model	TPR	FPR	Precision	F-Score	Class
LMT	0.992	0.005	0.998	0.995	Ham
	1.000	0.000	1.000	1.000	Norm
	0.995	0.008	0.982	0.988	Spam

From Table 1, it took LMT about 635.95 sec to create its model. The LMT has a classification accuracy of 99.305%. It is also clear from the results that LMT has performed excellently in term of very high correctly classified instances and very low number of incorrectly classified instances. The training and simulation error depicted in Table 2 shows that LMT produced an excellent classification result (0.993%) and very low error rate (0.0145). Once the model has been created, the next step is to analyse the results generated to determine the efficiency of the algorithms under consideration. Table 3 indicates that LMT have very good result in term of TPR, FTR, Precision and F-Score for ham, norm and spam classes. Below in Table 4 is the confusion matrices of the LMT algorithm which also provide a practical way for assessing the performance of the classifiers, each row of the table denotes actual rates of the class whereas each column indicates the predictions

Table 4. Confusion Matrix for LMT Algorithm.

Model	Ham	Norm	Spam	Class
LMT	3644	0	3	Ham
	0	8	0	Norm
	8	0	1,492	Spam

From the Table 4, LMT accurately predicts 3,644 instances out of 3,647 instances (3,644 ham instances that are truly ham and 8 spam instance that is really spam), and 3 instances wrongly predicted (3 instances of ham class predicted as spam). From our experiments it is clear that LMT performed excellently in term of effectiveness and efficiency considering its classification accuracy, TPR, FPR, precision and F-score. It also correctly predicts 1,492 instances out of 1,500 instances (1,492 spam instances that are truly spam and 8 ham instance that is really spam), and 3 instances wrongly predicted (3 instances of spam class predicted as ham).

CONCLUSION

Email is one the most predominant techniques for communication due of its inexpensive cost of sending messages, availability, ability to send and receive messages very fast. A good number of existing email spam filters cannot efficiently prevent spams from entering the user's inbox. This is due to the fact that spammers continue to devise more complicated methods for that can easily dodge spam filters.

In this study, we proposed Logistic Model Tree Induction algorithm for effective and efficient email spam filtering. And evaluated the performance of LMT algorithm on Enron spam datasets using accuracy, TPR, FPR, precision and F-measure to determine the algorithm with the accuracy. We conclude by from our study that LMT showed signs of future success as an algorithm that can be implemented either at mail server or at mail client side to further reduce the number of spam messages in email users inbox. In the future we will apply the algorithm on other publicly available email spam datasets to further evaluate its efficacy in filtering spams.

REFERENCES

1. Akshita T. 2016. "Content Based Spam Classification - A Deep Learning Approach". A Thesis Submitted to the Faculty of Graduate Studies, University of Calgary: Alberta, Canada.

2. Alkaht, I.J. and B. Al-Khatib. 2016. "Filtering SPAM Using Several Stages Neural Networks". *International Review on Computers and Software*. 11(2).

3. Asa Ben-Hur and J. Weston. 2010. "A User's Guide to Support Vector Machines". O. Carugo and F. Eisenhaber (eds.). *Data Mining Techniques for the Life Sciences, Methods in Molecular Biology*. 609, DOI 10.1007/978-1-60327-241-4_13, Humana Press, a part of Springer Science+Business Media, LLC 2010.

4. Awad, M. and M. Foqaha. 2016. "Email Spam Classification Using Hybrid Approach of RBF Neural Network and Particle Swarm Optimization". *International Journal of Network Security & Its Applications*. 8(4):17-28. DOI: 10.5121/ijnsa.2016.8402

5. Awad, W.A. and S.M. Elseuofi. 2011. "Machine Learning Methods for Spam E-mail Classification". *International Journal of Computer Science and Information Technology*. 3(1):173–184.

6. Bahgat, E.M., S. Rady and W/ Gad. 2016. "An E-mail Filtering Approach using Classification Techniques". In: *The 1st International Conference on Advanced Intelligent System and Informatics* (AISI2015), November 28-30, 2015. Springer International Publishing: BeniSuef, Egypt. 321-331.

7. Bouguila, N. and O. Amayri. 2009. "A Discrete Mixture-Based Kernel for SVMs: Application to Spam and Image Categorization". *Information Processing & Management*. 45(6):631-642.

8. Cao, Y., X. Liao, and Y. Li. 2004. 'An E-mail Filtering Approach using Neural Network". In: *International Symposium on Neural Networks*. Springer: Berlin, Germany. 688-694.

9. Christina, V., S. Karpagavalli, and G. Suganya. 2010. "Email Spam Filtering using Supervised Machine Learning Techniques". *International Journal on Computer Science and Engineering*. 2(09):3126-3129.

10. Dhanaraj, K.R. and V. Palaniswami. 2014. "Firefly and Bayes Classifier for Email Spam Classification in a Distributed Environment". *Australian Journal of Basic and Applied Sciences*. 8(17):118-130.

11. Fdez-Riverola, F., E.L. Iglesias, F. Diaz, J.R. Méndez, and J.M. Corchado. 2007. "Spam Hunting: An Instance-Based Reasoning System for Spam Labelling and Filtering". *Decision Support Systems*. 43(3):722-736.

12. Fonseca, D.M., O.H. Fazzion, E. Cunha, I. Las-Casas, P.D. Guedes, W. Meira, and M. Chaves. 2016. "Measuring Characterizing, and Avoiding Spam Traffic Costs". *IEEE Internet Computing*, 99.

13. Irwin, B. and B. Friedman. 2008. "Spam Construction Trends". In: *Information Security for South Africa* (ISSA). 1–12.

14. Karthika. R. and P. Visalakshi. 2015. "A Hybrid ACO Based Feature Selection Method for Email Spam Classification". *WSEAS Transaction on Computers*. 14:171-177.

15. Kaspersky Lab. 2017. "Spam Report". Visited on May 15, 2018. https://www.securelist.com/en/analysis/204792230/Spam_Report_April_2012, 2012.

16. Koprinska, I., J. Poon, J. Clark, and J. Chan. 2007. "Learning to Classify E-mail". *Information Sciences*. 177(10): 2167–2187.

17. Landwehr, N., M. Hall, and E. Frank. 2005. "Logistic Model Trees". *Machine Learning*. 59(1-2):161-205. https://doi.org/10.1007/s10994-005-0466-3.

18. Mason, S. 2003. “New Law Designed to Limit Amount of Spam in E-Mail”. http://www.wral.com/technolog

19. Savita, T. and B. Santoshkumar. 2014. “Effective Spam Detection Method for Email”. International Conference on Advances in Engineering & Technology - 2014 (ICAET-2014). *OSR Journal of Computer Science* (IOSR - JCE). 68-72.

20. Sharma, A. and A. Suryawans. 2016. “A Novel Method for Detecting Spam Email using KNN Classification with Spearman Correlation as Distance Measure”. *International Journal of Computer Applications*. 136(6):28-34.

21. Sosa, J.N. 2010. “Spam Classification using Machine Learning Techniques”. Sinespam. Master of Science Thesis. Master in Artificial Intelligence (UPC-URV-UB).

22. Sumner, M., E. Frank, and M. Hall. 2005. “Speeding up Logistic Model Tree Induction”. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer; 2005. 675-683.

23. Wang, X. 2005. “Learning to Classify Email: A Survey”. *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*.

24. Zavvar, M., M. Rezaei, and S. Garavand. 2016. “Email Spam Detection using Combination of Particle Swarm Optimization and Artificial Neural Network and Support Vector Machine”. *International Journal of Modern Education and Computer Science*. 68-74.

ABOUT THE AUTHORS

Dr. Emmanuel Gbenga Dada, holds a Ph.D. in Computer Science from the Department of Artificial Intelligence, Faculty of Computer Science and Information Technology at University of Malaya, Malaysia. He is a Senior Lecturer in the Department of Computer Engineering, University of Maiduguri, Nigeria. His current research interests include soft computing algorithms (particle swarm optimization, fuzzy logic, neural computing, evolutionary computation, machine learning, and probabilistic reasoning), swarm robotic, optimization techniques, and information security.

Dr. Joseph Stephen Bassi, received his Ph.D. degree in Electrical Engineering from the Universiti Teknologi Malaysia, in 2017, M.Eng. degree in Electrical & Electronics Engineering (Electronics) from University of Maiduguri, Nigeria in 2012 and B.Tech. degree in Computer Science & Mathematics from Federal University of Technology Minna, Nigeria in 2000. He is currently a Lecturer with the Department of Computer Engineering, Faculty of Engineering, University of Maiduguri, Nigeria. His research interests are in network algorithmic, data mining, artificial intelligence and optimization techniques, and computer communication networks.

SUGGESTED CITATION

Dada, E.G. and J.S. Bassi. 2018. **“**Logistic Model Tree Induction Machine Learning Technique for Email Spam Filtering”. *Pacific Journal of Science and Technology*. 19(2):96-102.