International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Model of Decision Tree for Email Classification

Nallamothu Naveen Kumar

Abstract: ***Aside from the undeniable benefits, the advent of the internet has resulted in a number of unfavourable security consequences. Spam emails are one of the most difficult problems that web users face. Spam refers to any and all emails containing unsolicited content that arrive in a user's email box. Spam can frequently cause network congestion and blocking, as well as system damage for receiving and sending electronic messages. As a result, distinguishing between spam and legitimate email has become critical. This paper presents a novel approach to feature selection and the Iterative Dichotomiser 3 (ID3) algorithm for generating a choice tree for email classification. The experimental results show that the proposed model achieves a very high level of accuracy.***

Keywords: Email Data Set, Data Pre-Processing, Extraction and Selection of Features, Decision Tree, Spam Detection

1. Introduction

The Internet, as a "network of networks, " has expanded communication and content placement options. The email system is one of the most efficient and widely used modes of communication [1]. Unfortunately, the steady increase in email users has resulted in a massive increase in spam emails. Spam emails are typically sent in bulk, with no regard for individual recipients. Spam emails, whether commercial or not, can cause serious problems in electronic communication. Spam emails generate a large amount of unsolicited data, affecting network bandwidth and storage capacity. Because of the large number of spam emails sent to email service users, it is difficult to distinguish between useful and unsolicited emails. As a result, managing and filtering emails is a significant challenge.

For spam detection, there are two main approaches. The first approach relies on email header analysis, while the second relies on email body analysis. Spam filters typically combine both approaches. Email header fields such as From, To, Subject, CC (Carbon Copy), and BCC (Blind Carbon Copy) almost always reveal the nature of the email. According to recent studies, the information provided by the email header is quite important [2], [3]. The assumption behind content-based filtering is that the body content of spam email differs from that of legitimate or ham mail. A variety of Machine Learning (ML) and data mining techniques have been used in recent years to classify email messages based on their content. To create an efficient email classifier, classification methods such as Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, and Neural Networks are commonly used [4]. The process of feature extraction and selection from email body is critical for most classification problems. Semantic properties of email content are used for feature selection and reduction in this paper. Various pre-processing steps, such as stop words removal, stemming, and term frequency, must be performed in order to detect spam emails efficiently [5], [6], [7]. The goal is to keep the most important features while reducing computation demand. Following feature selection, the ID3 algorithm generates a decision tree that classifies emails as spam or ham [8], [9]. The proposed method is evaluated in terms of accuracy, precision, and recall. The proposed system's performance is evaluated in relation to the size of the dataset and the size of the features.

The following is how this paper is structured. Section 2 describes in detail the proposed approach for spam detection. Section 3 summarises the findings, while Section 4 concludes.

2. SPAM Detection System

This section describes in detail the proposed Spam Detection (SD) system. The system is trained and tested in two stages. There are four modules in the training stage: data preparation, feature selection, feature reduction, and classification. Data preparation and classification modules are included in the testing stage.

1) E-mail dataset

The dataset used for classification contains 4000 entries [10]. There are 3465 ham messages and 535 spam messages in the dataset. This dataset is split into two parts: training and testing. The size of the dataset assigned for training purposes can affect system performance, as will be demonstrated later.

2) Dataset pre-processing

Before performing feature selection, the email dataset under consideration must be pre-processed. Spam emails are well known for containing phone numbers, emails, website URLs, money amounts, and a lot of whitespace and punctuation. Instead of removing the following terms, they are replaced with the following string for each training example:

a) Substitute 'emailaddr' for email addresses.
b) Substitute 'httpaddr' for URLs
c) Substitute 'moneysymb' for money symbols.
d) Substitute 'phonenumbr' for phone numbers.
e) Substitute 'number' for numbers.

Additionally, punctuation is dropped, and all white space including tabs, line breaks, and spacesis changed to a single space. A lowercase font was used across the entire dataset. Token words are used to separate the sentences. Tokenizing emails enables the detection of frequent spam terms. Likewise, stop words are taken out. Stop words are words like "a,” "an," "the,” and "is" that have no linguistic meaning. The subsequent step in the pre-processing stage is the stemming procedure. Stemming frequently entails the removal of derivational affixes and is described as "a rudimentary heuristic process that chops off the ends of words in the hope of most of the time attaining this goal properly" [11]. The pre-processing stage is critical because it narrows the search space for efficient feature extraction and selection.

3) Extraction and selection of features

The emails are analysed in this stage to determine which features (words) will be most useful in the classification stage. The main idea is to find words that occur frequently in the dataset or words that are relatively more important in understanding the class of an email. When determining whether an email is spam or not, the task is to determine whether there are any specific words or sequences of words that determine whether an email is spam or not. The Term Frequency (TF) method is used for this purpose. TF is a numerical statistic that is intended to reflect how important a word is to a document in a corpus. The TF value is proportional to how many times a word appears in a document. The size of a word is proportional to how frequently it appears in spam emails. Words with high TF weights, such as 'free, ' 'txt, ' and 'call, ' are good indicators of spam.

Text data is represented using the TF method for the ML algorithm. Because textual data is difficult to compute with, data representation is required. As a result, the frequency of all words in the pre-processed spam dataset is calculated, and the twenty most frequently occurring spam words are chosen as features. The occurrence of each feature in an email is then mapped in Table 1's feature matrix. One more feature has been added to improve the accuracy of the ML algorithm. This feature displays the total number of important spam words in an email. According to the experimental results, the corresponding feature has the greatest influence on the appropriate classification decision. In fact, for the majority of features, it is demonstrated that it is not important how many times a specific spam word appeared in an email, but rather whether it appeared at all. Because some features have no effect on the decision, this conclusion has enabled data dimensionality reduction. The feature that has no effect on class labels can be removed. Because of the feature reduction, the data is less sparse and statistically significant for the classification algorithm.

4) Decision tree

A decision tree employs a tree-like model to represent a variety of potential decision paths and their potential outcomes [13]. Each node in the decision tree represents a feature, each branch a decision, and each leaf an outcome (class or decision). By training a model on a set of labelled data, decision trees can be used to predict the class of an unknown query instance. Each training example should be distinguished by a number of descriptive characteristics or features. The characteristics can have nominal or continuous values. A decision tree is made up of root, internal, and leaf nodes. Internal nodes represent the conditions under which the tree divides into branches, while leaf nodes represent the possible outcomes for each path. Typically, each node has two or more nodes extending from it. "When classifying an unknown instance, the instance is routed down the tree based on the values of the attributes in the subsequent nodes, and when a leaf is reached, the instance is classified based on the class assigned to the leaf. " [14]. The main benefit of using a decision tree is that it is simple to follow and comprehend. Figure 1 depicts a typical decision tree. The words "free" and "money" are common spam words that are used as features. If the word "free" appears more than twice in an email, it is considered spam. Otherwise, we want to know if the email contains the word "money. " If the word "money" appears more than three times in an email, it is almost certainly spam; otherwise, it is ham.

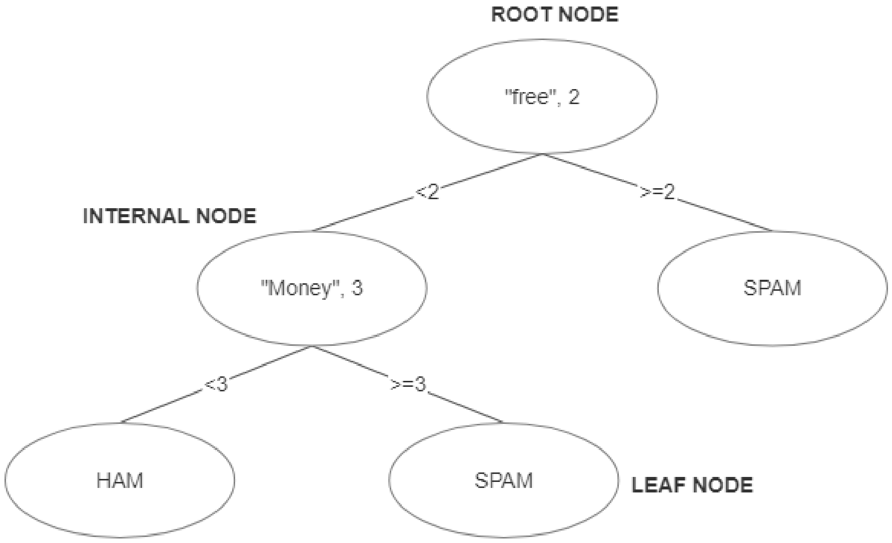


Figure 1. An Example of Decision Tree

The Decision tree algorithm serves as the foundation for the ID3 algorithm. The ID3 algorithm constructs the decision tree using entropy and information gain. "Entropy measures the impurity of a random sample collection, whereas information gain calculates the reduction in entropy by partitioning the sample based on a specific attribute" [15]. If the target attribute (class) has n different values, then the entropy S with respect to this n-wise classification is defined as follows:

Entropy (S)= $\sum_{i=1}^{n} -p_i \log_2 p_i$ (1)

where p. is the proportion/probability of S belonging to class Cn.

Information gain is calculated to split the attributes further in the tree. The attribute with the highest information gain is always preferred first. Entropy and information gain is related by (2):

gain (S, Ai)=Entropy (S)-Entropy$_{\mathrm{Ai}}$ (S) (2)

where EntropyA, {S) is the expected entropyif attribute

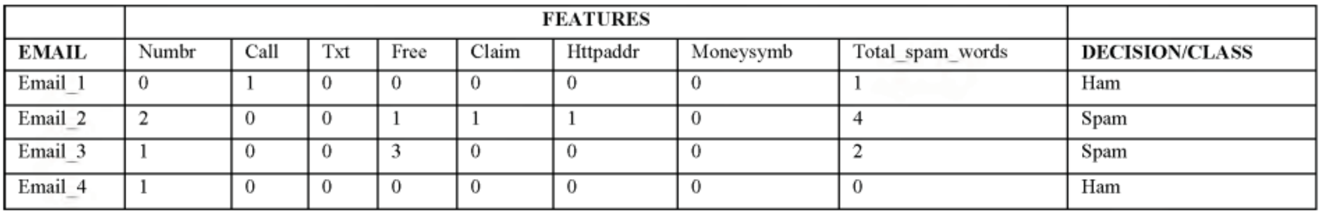
Ai is used to partition the data.

The algorithm was implemented in the following manner:

1) Establish a root node
2) Determine the entropy of the entire (sub) dataset.
3) Determine the information gain for each feature and choose the one with the highest information gain.
4) Assign the label of the feature with the greatest information gain to the (root) node. Grow an outgoing branch for each feature value and add unlabelled nodes at the end.
5) Split the dataset along the values of the maximum information gain feature and remove it.
6) Repeat steps 3-5 for each sub-dataset until a stopping criterion is met.

Because the chosen features have continuous values, converting continuous values to nominal values is required before performing a binary split. This is accomplished through the use of a threshold value. The threshold value is the value that provides the most information for that attribute. For the total spam words feature in Table 1, for example, the information gain is maximised when the threshold is set to two.

TABLE I
FEATURE MATRIX: EACH ROW REPRESENTS AN EMAIL WITH THE FEATURES PRESENTED IN COLUMNS



	FEATURES								
EMAIL	Numbr	Call	Txt	Free	Claim	Httpaddr	Moneysymb	Total_spam_words	DECISION/CLASS
Email_1	0	1	0	0	0	0	0	1	Ham
Email_2	2	0	0	1	1	1	0	4	Spam
Email_3	1	0	0	3	0	0	0	2	Spam
Email_4	1	0	0	0	0	0	0	0	Ham

3. Experimental Results

The proposed SD system's performance is measured using accuracy, prediction, and recall. A confusion matrix is created in order to compute these measures. The confusion matrix produces four results:

1) True Positive (TP): the number of instances that were correctly identified as spam.
2) True Negative (TN): the number of ham instances correctly identified.
3) False Positive (FP): the number of instances classified incorrectly as spam.
4) False Negative (FN): the number of instances classified incorrectly as ham.

The confusion matrix for email spam classification is shown in Table 2.

Table 2: Confusion Matrix

	Predicted HAM	Predicted SPAM
Actual HAM	True Negative	False Positive
Actual SPAM	False Negative	True Positive

Accuracy, precision, and recall can thus be defined as follows:

accuracy= $\frac{TP+TN}{TP+TN+FP+FN}$ (3)

precision= $\frac{TP}{TP+FP}$ (4)

recall= $\frac{TP}{TP+FN}$ (5)

For a classifier, accuracy is the proportion of total testing examples predicted correctly by the classifier, precision is the ratio of total number of correctly classified spam emails to total number of emails predicted as spam, and recall is the proportion of emails correctly classified as spam among all spam emails. The proposed SD system's performance is evaluated in relation to the size of the dataset and the size of the features. Table 3 shows the results.

Table 3: Classification Results based on Dataset Size and Feature Size

Dataset Size	Feature Size	Accuracy (%)	Precision (%)	Recall (%)
1000	7	97.4	92.01	87.21
1000	3	96.63	85.61	88.51
1500	7	97.32	92.28	86.21
1500	3	96.56	85.62	87.77
3000	7	97.2	91.52	85.71
3000	3	96.3	83.96	87.30

The performance is measured using datasets of various sizes. For example, with 1000 emails and 7 features used for training, the decision tree classifier achieved 97.4 percent accuracy. The precision and recall percentages are 92.01 and 87.21 percent, respectively. Reduced feature count reduces accuracy to 96.63 percent, with precision and recall values of 85.61 percent and 88.51 percent, respectively. The size of the dataset has a minor impact on accuracy: the accuracy for 1500 training examples and 3000 training examples was 97.32 percent and 97.2 percent, respectively.

4. Conclusion

In this paper, decision tree-based classification is used to detect spam emails. In addition, a novel approach to feature selection and reduction is presented. It is demonstrated that the system achieves high accuracy with only a few features and a small training dataset. It is planned to incorporate other classifiers and compare their performance with the proposed approach in the near future.

References

[1] P. Sharma and U. Bhardwaj, Machine Learning for Spam E-Mail Detection, International Journal of Intelligent Engineering & Systems, vol. 11, no. 3, 2017.

[2] A. S. Rajput, J. S. Sohal, V. Athavale, “Email Header Feature Extraction using Adaptive and Collaborative approach for Email Classification”, in International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, vol.8, Issue7S, May2019

[3] P. Kulkarni, J. R. Saini and H. Acharya, “Effect of Header-based Features on Accuracy of Classifiers for Spam Email Classification”, in: International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 3, 2020

[4] E. G. Dada, S. B. Joseph, H. Chiroma, S. Abdulhamid, A. Adetunmbi, E. Opeyemi and Ajibuwa, “Machine learning for email spam filtering: review, approaches and openresearch problems”. in Heliyon, June 2019

[5] E. M. Bahgat, S. Rady, W. Gad and I. F. Moawad, “Efficient email classification approach based on semantic methods”, In: Ain Shams E n g. J., vol. 9, no. 4, pp. 3259-3269, December 2018.

[6] F. Ruskanda, “Study on the Effect of Preprocessing Methods for Spam Email Detection”, in: Indonesian Journalon Computing (Indo-JC). 4. 109, March 2019.

[7] A. Sharma, Manisha, D. Manisha and D. R. Jain, “Data Pre-Processing in Spam Detection”, in: International Journal of Science Technology and Engineering (IJSTE), vol. 1, Issue 11, May 2015.

[8] L. Shi, Q. Wang, X. Ma, M. Weng and H. Qiao, “Spam Email Classification Using Decision Tree Ensemble”, in Journal of Computational Information Systems 8, March 2012.

[9] S. Balamurugan and R. Rajaram, “Suspicious E-mail Detection via Decision Tree: A Data Mining Approach”, January 2007.

[10] T. A. Almeida and J. M. Gomez Hidalgo, SMS Spam Collection, UCIMachine Learning Repository, viewed 12 September 2020, https://archive. ics. uci. edu/ml/datasets/sms+spam+collection

[11] C. D. Manning, P. Raghavan and H. Schutze, “Introduction to Information Retrieval”, in Cambridge University Press, 2008.

[12] A. Bhowmick and S. M. Hazarika, “Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends”, 2016.

[13] J. Grus, “Data Science from Scratch: First Principles with Python”, O'Reilly Media. Inc., April 2015.

[14] I. H. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations”, Morgan Kaufmann, San Francisco, 2000.

[15] T. Kristensen and G. Kumar, “Entropy based disease classification of proteomic mass spectrometry data of the human serum by a support vector machine”, Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.