International Journal of applied exercise physiology

ISSN: 2322-3537
Vol-14 Issue-01 June 2025

Using a Decision Tree Model to Sort Emails

[1]Manjusha Nambair, [2] M. Nikitha,

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract—

The expansion of the Internet has brought about several negative security consequences, in addition to its obvious advantages. One of the biggest problems that people have while using the Internet is spam emails. Any unsolicited email that ends up in a user's inbox is considered spam. In many cases, spam may cause electronic message delivery systems to become overloaded, stop working, or even crash. This highlights the critical importance of accurately distinguishing spam from authentic email. In order to create the decision tree for email classification, this study introduces a novel method for feature selection using the Iterative Dichotomiser 3 (ID3) algorithm. The experimental findings show that the suggested model accomplishes an exceptionally high level of accuracy.

Introduction

The Internet's role as a "network of networks" has greatly increased the reach and accessibility of information. Among the most popular and efficient means of communication, the email system ranks high [1]. The exponential growth of email users has, however, coincided with a corresponding explosion in spam. Rather of addressing each recipient individually, spam emails are typically sent in bulk. Electronic communication may be severely disrupted by spam emails, regardless of their commercial character. The storage and network capacity are also impacted by the massive amounts of unwanted data produced by spam emails. Users of email providers have a hard time differentiating between legitimate and spammy emails because of the sheer volume of them. Consequently, a significant obstacle is the management and filtering of emails. The goal of the filtering process is to identify and remove unwanted emails. For the purpose of detecting spam, two primary methods exist. The first method relies on analyzing email headers, whereas the second method uses email bodies as its foundation. Both of those methods are often used together by spam filters. Fields such as From, To, Subject, CC (Carbon Copy), and BCC (Blind Carbon Copy) in an email's header practically disclose the email's content. The data supplied by an email's header is crucial, according to recent research [2, 3]. The premise upon which content-based filtering is built is that spam and valid email have distinct body contents. Several data mining and Machine Learning (ML) methods have been used for email content classification in the last several years. Efficient email classifiers are often developed using classification techniques as Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, and Neural Networks [4]. To address the majority of categorization issues It is crucial to extract and pick features from the email content. Feature selection and reduction in email content is accomplished in this research by using its semantic features. Stop word removal, stemming, and term frequency are a few of the preprocessing procedures that must be performed in order for spam email detection to be effective [5, 6, 7]. The goal is to minimize computation requirements while preserving the most critical characteristics. A decision tree is generated using the ID3 algorithm to classify emails as spam or ham after feature selection [8], [9]. Accuracy, precision, and recall are used to assess the suggested technique. Dataset and feature size are used as metrics to evaluate the proposed system's performance. In this article, the following structure is used. The suggested method for spam detection is described in depth in Section II. Section IV provides the conclusion, whereas Section III provides a summary of the findings.

SPAM DETECTION SYSTEM

A detailed presentation of the proposed Spam Detection (SD) system is provided in this section.

International Journal of applied exercise physiology
ISSN: 2322-3537
Vol-14 Issue-01 June 2025

Training and testing are the two phases that the system goes through. Four steps make up the training phase: preparing the data, selecting features, reducing features, and classification. Modules for data preparation and classification make up the testing phase. The SD process is shown in Figure 1, and the suggested method is described in sections that follow.

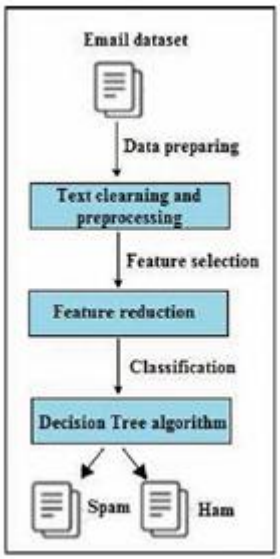

Figure 1. Spam detection process

Message database Four thousand items make up the dataset used for the classification purpose [10]. In the dataset, you may find 3,465 legitimate mails and 535 spam reports. The two halves of this dataset are the training set and the testing set. As we'll see later on, the amount of the dataset used for training might impact the performance of the system. Section B: Dataset Preprocessing It is necessary to preprocess the email dataset in question prior to feature selection. It is well recognized that the majority of spam emails include contact information, URLs to websites, financial sums, and an abundance of punctuation and whitespace. Rather of deleting them, the following phrases are substituted with a unique string for every training example: 1. 2. 3. 4. Substitute "emailaddr" with actual email addresses. Swap out URLs for "httpaddr" Switch out the money symbols for'moneysymb' Substitute "phonenumbr" for actual phone numbers. 5. Substitute "number" for all numbers. Aside from erasing punctuation, the text also has all whitespaces (spaces, line breaks, tabs) replaced with a single space. All of the data is in lowercase as well. Tokens are used to break down the phrases into individual words. In order to identify common spam terms, every email is tokenized. Termination terms are also eliminated. Certain words, such as "a," "an," "the," "is," etc., do not possess any grammatical significance and are hence considered stop words. Stemming is the subsequent stage of preprocessing. In an effort to achieve this aim consistently, stemming often involves removing derivational affixes and is generally described as a rudimentary heuristic procedure [11]. In order to efficiently extract and pick features, the preprocessing step is crucial since it narrows the search area. C. Extracting and selecting features This step involves analyzing the emails to identify the terms and traits that will be most helpful throughout the categorization process. Identifying frequently recurring terms in the dataset or words with substantially greater value in determining an email's class is the key premise. It is important to know if there are any particular terms or sequences of words that may be used to detect spam emails. The Term Frequency (TF) technique is used for this objective. The term "TF" refers to a numerical statistic that attempts to represent the importance of a word in relation to a document in a corpus. The frequency with which a word occurs in a given text has a direct correlation to its TF value. A word cloud of frequently used terms in spam emails is shown in Figure 2. Figure 2 shows that the frequency of a term in spam emails is directly correlated with its size. Common spam signs include the words "free," "txt," and "call," all of which have high TF weights.

Figure 2. Visual representation of important words for spam email

Machine learning algorithms employ the TF technique to represent text data. Textual data makes calculation difficult, hence data representation is

International Journal of applied exercise physiology
ISSN: 2322-3537
Vol-14 Issue-01 June 2025

necessary. As a result, we rank the terms in the preprocessed spam dataset by frequency and use the top twenty as features. Table 1 shows the feature matrix that is used to map the presence of each characteristic in an email. An additional feature is included to improve the accuracy of the ML algorithm. That characteristic stands for the sum of all the significant spam terms found in a given email. From what we can tell from the experiments, the corresponding trait is the one that matters most for making the right categorization call. Indeed, it has been shown that for the majority of characteristics, the frequency with which a particular spam phrase occurs in an email is less essential than its mere presence. Data dimensionality reduction has been made possible by this result, since some attributes do not impact the choice. Toss out the feature that doesn't affect the class labels. Less sparse and more statistically meaningful data for the classification algorithm has resulted from feature reduction.

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

Preference tree A decision tree is a model that employs a tree-like structure to show many option routes and the consequences that might follow from them [13]. A characteristic is represented by a node in a choices tree, a decision by a branch, and a result (class or decision) by a leaf. The class of an unknown query instance may be predicted using decision trees by training a model on a collection of labelled data. There should be a number of descriptive qualities or properties that each training sample has. The characteristics' values might be nominal or continuous. There are three types of nodes in a decision tree: root, internal, and leaf. The tree's internal nodes stand for the factors that cause it to branch out, while the leaf nodes indicate the potential consequences of each branch. In a normal network, two or more nodes branch out from every given node. To classify an unknown instance, we first use the values of its attributes at each node to guide its path down the tree; once we reach a leaf, we categorize the instance based on the class associated with that leaf. cited as [14]. The clarity and simplicity of the decision tree structure is its primary strength. An example of a standard decision tree is shown in Figure 4. As features, the phrases "free" and "money" are common in spam. Emails are considered spam if they include the term "free" more than twice. Aside from that, we are inquiring about the presence of the term "money" in the email. Emails with more than three instances of the word "money" are almost sure spam; those without are likely ham.

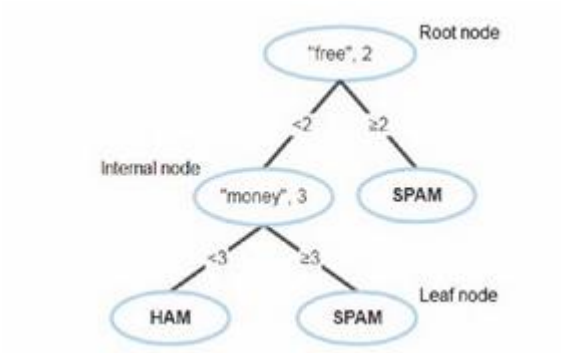

Figure 3. An example of decision tree

A decision tree algorithm is the basis of the ID3 algorithm. The decision tree is constructed using the ID3 algorithm's entropy and information gain metric. "The information gain determines the reduction in entropy by partitioning the sample according to a certain attribute, while entropy measures the impurity of an arbitrary collection of samples" [15]. The entropy S with respect to this n-wise classification may be described as (1) if the target characteristic (class) can take on n alternative values:

International Journal of applied
ISSN: 2322-3537
Vol-14 Issue-01 June 2025

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

The calculation of information gain is used to further separate the characteristics in the tree. Prioritization is always given to the characteristic that yields the most new information. Here is a relationship between entropy and information gain:

$$gain(S,A_i) = Entropy(S) - Entropy_{A_i}(S) \quad (2)$$

where the predicted entropy, denoted as EntropyA,{S), is calculated when attribute Ai is used for data partitioning. The following steps were followed to implement the algorithm: 1. Establish a primary node 2. 3. 4. 5. 6. Determine the total (sub)dataset's entropy. Determine which characteristic provided the most useful information by calculating its information gain. Put the label of the feature that yields the most information at the (root) node. Extend an outgoing branch for every feature value and terminate it with unlabeled nodes. Divide the data set along the greatest information gain feature's values and then delete it. Continue with each sub-dataset as before, repeating steps 3–5 until a stopping criterion is met. To execute a binary split, it is necessary to transform continuous values into nominal ones, as the selected features possess continuous values. The threshold value is used for that purpose. For any given property, the optimal threshold value is the one that yields the most useful information. For the total spam words feature in Table 1, for instance, the optimal threshold is two, which maximizes the information gain.

EXPERIMENTAL RESULTS

Accuracy, prediction, and recall are used to assess the performance of the proposed SD system. A confusion matrix is constructed to calculate these metrics. Four results are produced by the confusion matrix: 1. The amount of occurrences that were accurately identified as spam is known as True Positive (TP). 3. The number of cases that were properly identified as ham is known as the True Negative (TN). The amount of occurrences that were mistakenly labeled as spam is known as the False Positive (FP). 4. The number of cases that were mistakenly labeled as ham is known as a False Negative (FN). A confusion matrix for email spam categorization is shown in Table 2.

TABLE II CONFUSION MATRIX

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

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

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

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

The accuracy, precision, and recall of a classifier are defined as follows: accuracy, which is the percentage of testing examples that the classifier correctly predicted, recall, which is the percentage of spam emails that were correctly classified, and total number of emails that were predicted as spam. The dataset and feature sizes are used as metrics to evaluate the suggested SD system's performance. In Table 3 you can see the outcomes.

TABLE III 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

Performance is evaluated using datasets of varying sizes. As an example, a decision tree classifier achieved an accuracy of 97.4 percent when trained using 1000 emails and seven characteristics. A recall of 87.21% and an accuracy of 92.01% are recorded. With fewer characteristics, accuracy drops to 96.63%, while recall and precision drop to 85.51% and

International Journal of applied exercise physiology
ISSN: 2322-3537
Vol-14 Issue-01 June 2025

88.51%, respectively. Accuracy is marginally affected by dataset size; for instance, 97.2% accuracy was achieved with 3000 training instances, but 97.32% accuracy was achieved with 1500 examples.

CONCLUSION

This article uses spam email detection using decision tree-based categorization. Also included is a whole new method for narrowing down features. The method is shown to attain impressive accuracy using just a modest number of characteristics and a very short training dataset. We want to implement more classifiers and evaluate their effectiveness in comparison to the suggested method soon.

REFERENCES

[1]. [1] P. Sharma and U. Bhardwaj, Machine Learning based Spam E-Mail Detection, in International Journal o f Intelligent Engineering & Systems, vol. 11, no. 3, 2017 A. S. Rajput, J. S. Sohal, V. Athavale, "Email Header Feature Extraction using Adaptive and Collaborative approach for Email Classification", in International Journal o f Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, vol.8, Issue 7S, May 2019

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

[3]. 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 open research problems". in Heliyon, June 2019

[4]. E. M. Bahgat, S. Rady, W. Gad and I. F. Moawad, "Efficient email classification approach based on semantic methods", In: A in Shams Eng. J., vol. 9, no. 4, pp. 3259-3269, December 2018.

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

[6]. A. Sharma, Manisha, D. Manisha and D.R. Jain, "Data Pre Processing in Spam Detection", in: International Journal o f Science Technology & Engineering (IJSTE) , vol. 1, Issue 11, May 2015

[7]. L. Shi, Q. Wang, X. Ma, M. Weng and H. Qiao, "Spam Email Classification Using Decision Tree Ensemble",in Journal o f Computational Information Systems 8, March 2012 S.

[8]. Balamurugan and R. Rajaram, "Suspicious E-mail Detection via Decision Tree: A Data Mining Approach", January 2007. [10]

[9]. 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

[10]. [11] C. D. Manning, P. Raghavan and H. Schutze, "Introduction to Information Retrieval", in Cambridge University Press, 2008.

[11]. A. Bhowmick and S. M. Hazarika, "Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends", 2016 [13]

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

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

[14]. [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