WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Conceptualization and Analysis of Junk Mail Determination System

Gaurav Dahiya [a] ***Shivkant*** [b]

[a] *Student, Sat Kabir Institute of Technology and Management, Bahadurgarh, India*
[b] *Assistant professor, Sat Kabir Institute of Technology and Management, Bahadurgarh, India*

A B S T R A C T

In today's computerised business environment, information sharing throughout organisational divisions is essential. Email is a crucial and important tool for quick and inexpensive communication. People may communicate with one another simply thanks to e-mails. However, the convenience of email also comes with some drawbacks, such as junk emails (or unwanted bulk email). These junk messages have turned into a problem for businesses and the IT sector. This rapid growth of junk mail, which are superfluous and overwhelm some important emails, uses up memory and bandwidth, and it takes time to delete them. In this investigation, we focused on the issue of junk email, that has become a significant worry in the online community. The creation of a reliable anti-junk filter was required by the rising volume of unwanted emails (junk mails). With the use of machine learning (ML) techniques, junk e-mail is now successfully filtered on a constant basis. In this study, we examined a few well-known ML strategies and how well they applied to the challenge of categorising junk mail. We have created descriptions of the procedures and the variations in how they are carried out based on the amount of Junk Assassin users. The present paper proposes a system for screening junk mail based on content, using a mix of novel and established techniques. The emails' features have been retrieved. The emails have been preprocessed, and a vocabulary list has been made. The datasets of the emails were trained using a support vector machine (SVM) machine learning classifier. After that, we divided the emails into junk and non-junk categories. Every test was run on the well-known, publicly available corpora.

Keywords: Junk Mail, Primary Mail, Support Vector Machine

1. Introduction

Email is a popular and widely utilised mode of communication today. The vast user base, estimated to be close to billions of individuals, and the daily increase in users can be used to explain the communication channel's incredibly quick uptake (Email, 2005) The small advantage of this communication method is that it is practically prompt, impulsive, and inexpensive. The email system typically uses the SMTP protocol. The existing email service is vulnerable to exploitation because of how straightforward it is. The techniques were developed with the assumption that email users wouldn't make use of the capability to send texts to one another. The inappropriate and excessive use of the messaging feature has evolved through time and taken many different shapes. Regular instances of misuse include forged emails, unsolicited emails (junk), dishonest behaviour, and identification theft via "phishing" emails. Abuse instances involve email denial-of-service attacks, virus and worm attachments, and more. All of these types share the same trait: they all profit from the lack of sender and recipient controls and authentication in the email system. Whomever can send a message to anyone via email without needing their permission in advance.

In our work, we have focused on the problem of junk mails which are received in bulk by every user on a daily basis. The junk mails are the undesirable mails delivered by any unsolicited person also termed as junkmer (yang, 2011) with the purpose of producing wealth. Users spend the majority of their time (Kumar, 2015) deleting junk mails from their inboxes. Many replicas of the similar message are delivered on multiple occasions, which not only costs money to an organization (DíAz, 2012), but also frustrate the recipient. Junk emails invade users' inboxes and produce a large amount of unneeded data, which lowers network capacity and usage. In order to separate email data into junk and genuine emails, this study suggests using a Junk Mail Detection (SMD) system. The email address, subject, and content of the message are the three main factors in junk screening (Sharma, 2014).

The subject and content of an email are two parts that are present in all emails. A junk email's content can be filtered to help categorise it. The idea that the subject matter of junk mail differs from that of legitimate or junk mail is the basis for its identification. Words connected to commercial adverts, service recommendations, dating-related content, etc. Junk email detection techniques fall into two categories: knowledge engineering-based methods and machine learning methods (Ma, 2019). Network-based email classification using knowledge engineering considers IP (Internet Protocol) addresses, network addresses, and a specified set of rules.

The procedure is time-consuming, despite the excellent outcomes it has produced. The process of maintaining and altering rules can be inconvenient for some users. Expert systems are less productive than machine learning, which does not need any rules (Guzella, 2009). Based on its material and other factors, the classification system assigns a categorization to the email. For the majority of classification problems, the process of feature extraction and selection is crucial. Features play a significant role in the classification procedure. In this research, a CFS (Mohamad, 2015) technique is used for feature extraction. The CFS methodology selects the best features from a set of features for efficient classification outcomes. The proposed Junk mail detection

(JMD) method includes a special mix bagging procedure to overcome the drawbacks of the traditional model. In Fig. 1, the fundamental process of email filtering is shown.

2. Related Work

The two most common techniques for junk mail filtering are knowledge engineering (KE) and machine learning. We provided a set of criteria in KE-based ways to distinguish between legitimate emails and junk emails. The customer or the software vendor who provides a particular rule-cantered junk straining programme may specify this set of rules. It is always impossible or potentially inconvenient for many people to regularly renew and maintain these norms, which is necessary for success. It was determined that machine learning (ML) methods are superior to knowledge engineering methods since they do not require any instructions to be specified after numerous successful ML approaches.

Instead, a set of training models—which are a collection of emails with certain subject lines—are used in machine learning techniques. Following that, a specific algorithm was used to extract the categorization rules from these email posts. Different machine learning techniques, including artificial immune systems, J48 classifiers, SVMs, Naive Bayes, and neural networks, have recently received a lot of attention and can be applied to e-mail filtering. The analysis filtering of junk mail and the protection of regular emails were the main foci of this study. Email must be filtered in order to be divided into primary and junk. Through the identification of distinctive qualities, the authors (Mohamad, 2015) have presented a junk email filtration approach that divides emails into legitimate and unwanted ones. They employed TF-IDF and rough set theoretical technique after pre-processing the dataset's (English and Malay email) characteristics. They then used a machine learning strategy to categorise data and achieved good performance.

A new ML-centered technique has been suggested by authors in (Harisinghaney, 2014) for the classification of email data. The algorithmic implementation includes KNN and Naive Bayes algorithms and presents practical results in cases where algorithms are used to pre-processed datasets. Additionally, authors in (Youn, 2007) have designed an email filtering strategy that is centred on ontologies. The J48 decision tree-centered method was utilised to categorise the used dataset. A RDF linguistic-centered ontology was created by Jena in order to test the results obtained after categorization.

The authors of (Faris, 2015) employed the FFN network focused technique to pinpoint the results and identify junk emails. The dataset, which is uniformly divided into two divides for training and testing purposes, was trained using the Krill Herd algorithm. The authors of (Caminhas, 2000) have outlined a comprehensive study of recent advancements in the use of ML techniques for junk filtering. They have placed a strong emphasis on both text- and image-centered approaches. Instead than seeing Junk filtering as a conventional classification problem, they have emphasised the importance of reflecting specific aspects of the issue, particularly perception meaning for the construction of various filters.

3. Methodology

3.1 Junk mail filtration process

There are several intermediary processes that are completed during pre-processing and post-processing that are part of the categorization process. In order to extract the most useful information from a collection of email files, processes including tokenization, lemmatization, and stop word removal are performed when an email file reaches the junk filter. These actions are classified as pre-processing. In the representation step, a structure between the email files and associated features that were obtained during pre-processing is also being determined.

The relationship between the features and email files needed to train the classifier mask image and then determine the boundary of the target region is fundamentally represented by this structure. Post-processing is where this stage is completed (Fig. 2). Classification is the process of choosing the best class label for a given input. In simple classification tasks, each input is handled independently from every other input, and the set of labels is predefined. Here are a few examples of categorising tasks: determining whether a message is junk. selecting a topic for a news article from a predetermined list of alternatives like "sports," "technology," and "politics." figuring out whether the word "bank" is used to describe a river bank, a financial institution, a tilting motion, or the act of depositing cash in a bank. A header and body are the two sections of an email.

3.2 Formation of an email junk classifier

It is difficult to distinguish between Primary (real emails) and Junk because of the complexity that junkmers provide, this has prompted interest in research into Junk Classification (spontaneous emails). Both sides are engaged in this conflict. Junkmers develop new attacks to work around newly found filtering techniques. The framework for an email classifier is shown in Fig. 3.

3.3 Proposed work

The JunkAssassin corpus was used for our research studies. This corpus includes some recent and earlier junk that wasn't produced by junk traps. 2300 junk mail files were chosen from the entire corpus for this study. Additionally, this bundle contains a mix of straightforward and intricate Genuine (Primary) files that together create 2300 Primary email files. Simple primary emails in this corpus may be identified and categorised, however it is challenging to quantify complicated primary emails. Complex primary includes specific attacks from junkmers that make it very important to distinguish emails, such as the use of HTML, unusual HTML markup, coloured script, and commentary on the features.

The data set is divided into two groups: a training set and a test set, respectively, each of which contains 702 and 260 emails that are equally split between junk and primary mails. Any text mining task begins with text purification, which involves removing words from the text that might not be relevant to the data we are trying to extract. Junk may be difficult to identify in emails because they frequently contain unwelcome elements like punctuation, stop words, digits, and other symbols.

Stop words such "and," "the," "of," and "other" are common in all English phrases but serve no purpose in evaluating whether an email is junk or real. As a result, they have been removed from the emails. The process of lemmatization combines a word's numerous inflected forms so they can be studied as a single entity. The word "include" is used to describe the words "include," "includes," and "included." Lemmatization, as opposed to stemming, keeps the sentence's context. Punctuation and other non-word symbols, as well as special characters, must still be eliminated from the postal documents. There are numerous ways to accomplish this. We will eliminate such words after creating a dictionary, which is a very practical approach because, with a dictionary, you only need to eliminate such words once.

3.4 Making word dictionary

The first line of the email contains the subject, and the third line contains the email body. We will just employ text analytics in this case to find junk emails. We must first create a list of words and the instances in which they occur. 700 emails from a training collection are used for this. After the dictionary has been created, we can modify the aforementioned function by adding a few lines of code to remove the non-words we discussed in step 1 of the process.

3.5 Feature extraction process

Once the vocabulary is complete, we can extract a word count vector (our feature in this case) with 3000 dimensions for each email in the training group. The 3000 words in the training file are represented by each word count vector. Actually, I think you've guessed that most of them will be zero. Let's examine a case in point. Let's say we have 500 terms in our vocabulary. In this file, the frequency of 500 dictionary words is recorded for each word count vector used in the training.

3.6 Training the classifiers

The classifier for Support Vector Machines (SVM) has been trained. supervised binary classifiers called SVMs are useful when there are numerous features to take into account. From the rest of the data (the boundary of the separating hyper-plane), support vectors, a subset of the training data, are extracted using SVM. Based on support vectors and a kernel technique, the SVM model's decision function predicts the class of the test data. After the classifiers have been trained, we may evaluate how well the models perform on a test set. We obtain the word count vector for each email in the test set and forecast whether it belongs in the primary or junk category using the trained SVM model. The primary concept of an SVM is to develop a nonlinear kernel function to transfer data from the input space to a possibly high feature space, and then generalise the ideal hyper-plane with the greatest difference between the two categories.

Only a small portion of the training set, also known as the support set, and its constituents, the support vectors, contain non-zero values for the multipliers ai. SVMs are strong candidates for rigorous or discriminating sampling approaches that look for these trends since they construct a hypothesis using a subset of the data containing the most illuminating contours. If the data were initially unlabeled, a good heuristic algorithm would ask for labels for the patterns that would make support vectors. When classifying data is expensive or the dataset is large and unlabeled, active selection would be extremely helpful.

As with any supervised learning model, the classifier is initially cross-checked once an SVM has been trained. Utilise the computer that has been taught to categorise (predict) new data. To obtain appropriate predicted accuracy, we can also use a number of SVM kernel functions, however we must change the kernel functions' constraints (Jayant, 2021).

3.7 Functioning of SVM

The 2Dimension sample dataset, which is divided by a linear border, is where we started. The plotting of the training data is shown in Figure 4. A clear difference has been made (denoted by o) between the locations of positive examples (represented by a +) and negative instances (marked by a -). These two SVM decision boundaries differ from one another in this manner. With SVM, we may make use of different C parameter values. A high C value indicates that SVM is making an effort to accurately classify every case.

C is the same as the regularisation parameter for logistic regression ().When C=1, the SVM incorrectly classifies the far-left data point by positioning the decision border at the intersection of the two datasets (Figure 5). All cases are accurately classified by the SVM if C=100, but the decision boundary does not appear to be truly ideal for the data.

SVMs have been utilised in this experiment to achieve non-linear classification. Non-linear separable dataset is being used here. To use the SVM to define non-linear decision limits, we should develop a Gaussian kernel. Let's think of the Gaussian kernel as a similarity function that determines how far apart two examples, (x(i); x(j)) are from one another.

Figure 6 now displays the SVM classifier's load, plot of data set 2, and results. This time, the +ive and -ive examples of data set two do not have a linear separation boundary. However, if we combine the SVM with the Gaussian kernel, we can create a non-linear decision boundary that will work quite well for dataset 2. Figure 7 demonstrates that we have a distinct decision boundary with the aid of the Gaussian kernel function. The decision boundary is capable of accurately tracking the dataset's contours and separating the majority of the +ive and -ive cases.

The majority of current email systems offer a system or filter that can distinguish between junk and legitimate emails. Here, we've created our own junk filter using an SVM classifier. We have built a classifier that can distinguish between junk and junk mail (y = 0 and y = 1). Each email has been transformed into a feature vector (x Rn). We used Dataset, a subset of the JunkAssassin public corpus. We will only use the body of the email for feature extraction and classification because we have constructed a content-based filter.

Let's look at an example email, like the one in figure 8. The scenario that came before has a URL, an email address (in the final), numbers, and dollar amounts. While different emails will contain the same types of information (such as numbers, URLs, or email addresses), the specific items (such as the actual URL or dollar amount) in almost every email will be different. As a result, standardising these values is a frequent method for processing emails, ensuring that all URLs, integers, and other variables are handled uniformly. For instance, we may replace every URL in the email with the special string httpaddr in order to "indicate the presence of a URL."

This enables the junk classifier to categorise messages based on the presence or absence of URLs rather than on a specific URL. Since junkmers frequently randomise URLs, it is extremely unlikely to find the same URL in a brand-new junk message. The procedures for processing and normalising emails are listed below.

After completing the operation of email preprocessing, we have a word list for each email. Additionally, we will choose the words we want to exclude from our classifier as well as the ones we want to include. We used the words that were used the most frequently to create the vocabulary list. The following is an illustration of a vocabulary list (figure 9). The terms in the junk corpus that appear at least 100 times in our vocabulary list were chosen, resulting in a list of 1900 words.

A vocabulary list of 10,000 to 50,000 words has typically been used. Now that we have a vocabulary list, we can map each word in the precompiled emails into a list of word indices that contains the index of the word in the vocabulary list. In the sample email, the word "anyone" was first converted to "anyon" and then mapped to index 86 (highlighted) in the vocabulary list.

The feature extraction process, which converts each email into a vector in Rn, has been added next. An email's featurexi 0|1; indicates if the i-th word from the dictionary is present. In other words, if the email contains the i-th word, then xi = 1; if it does not, then xi = 0.

4. Results and Detections

A preprocess dataset that is used to train the SVM classifier has been loaded. Each email has been analysed, and features have been extracted and generated into feature vectors. As can be observed, after training, the classifier has a training precision of almost 99.8% and a test correctness of approximately 98.50%. To further understand how it operates, we may look at the parameters to find out which phrases the junk classifier considers to be the most analytical of junk. The words that correspond to the classifier's parameters with the highest positive values are also displayed. As a result, it is likely that an email including the words "guarantee," "delete," "dollar," and "price" will be labelled as junk. Look at figure 10. In light of these findings, we can finally state that processed email is junk email (figure 11).

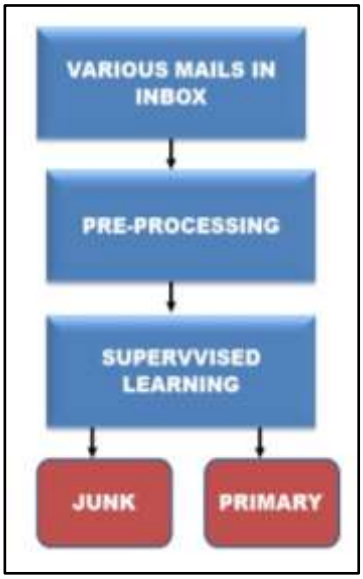

Fig. 1: A simple junk Mail Labelling

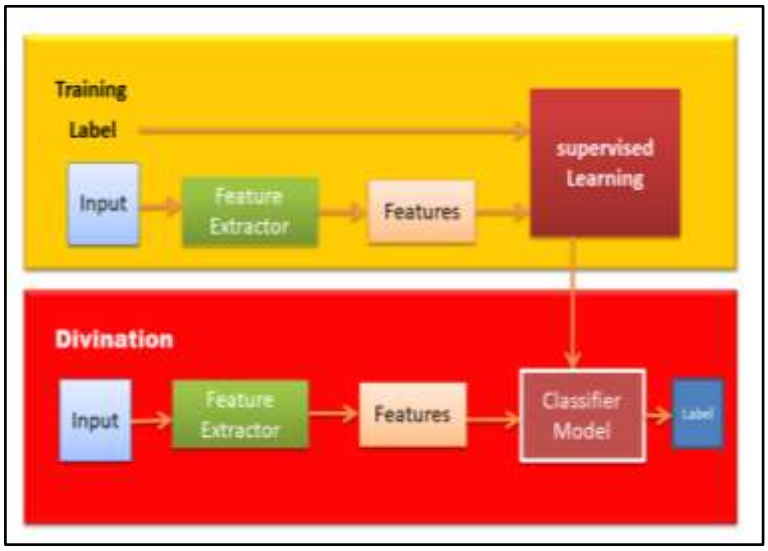

Fig. 2: Email files and feature relationships for classifier training

5. Conclusion

In this study, we explored a few well-known ML approaches and their applicability to the problem of categorising junk e-mail. The experiment in this study will focus on the email body because content-based machine learning techniques have adopted it widely. Several feature selection and feature search approaches have been reported for the pre-processing in order to choose the most useful characteristics. Comparative research on feature selection and feature search methods have been carried out separately and are addressed in the following chapters in order to determine the most informative characteristics. The email has been preprocessed, and a vocabulary list has been created. With this list, we have matched the word indices. After that, we finished the feature vector and did feature selection. After that, the SVM classifier was trained. We've come up with a set of words. We determine whether a word is junk or primary. Our findings indicate a 98% accuracy.

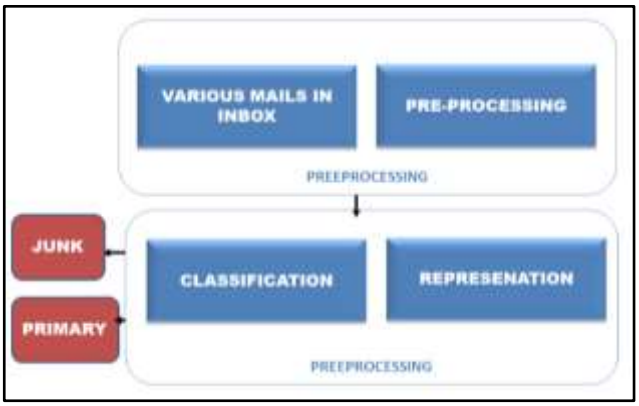

Fig. 3: Classification of E-mails

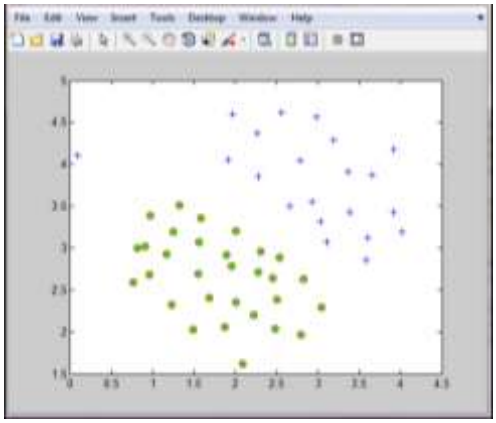

Fig. 4: Sample of E-Mails

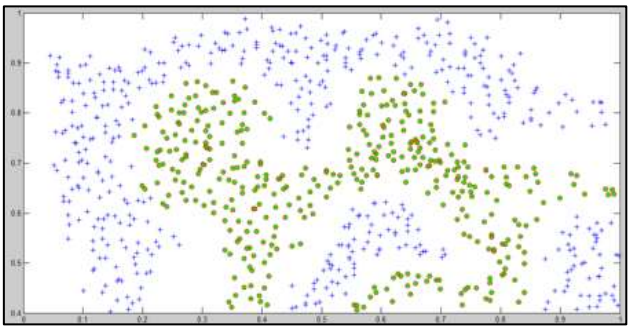

Fig. 5: Training Samples

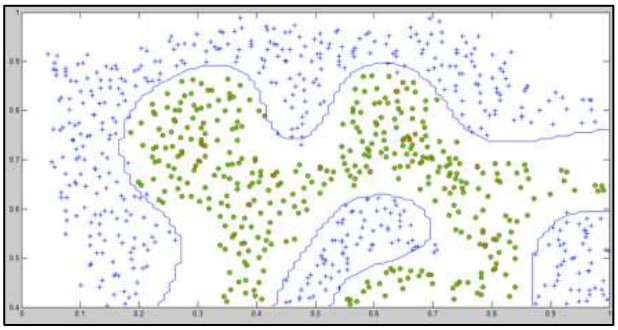

Fig. 6: Decision Barrier for set 2 using SVM and Gaussian Kernels

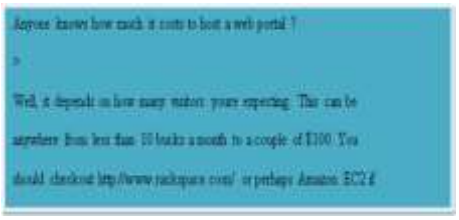

Fig 7: Taster E-mail

Fig 8: Pre-processed E-mail

Fig 9: Vocabulary list

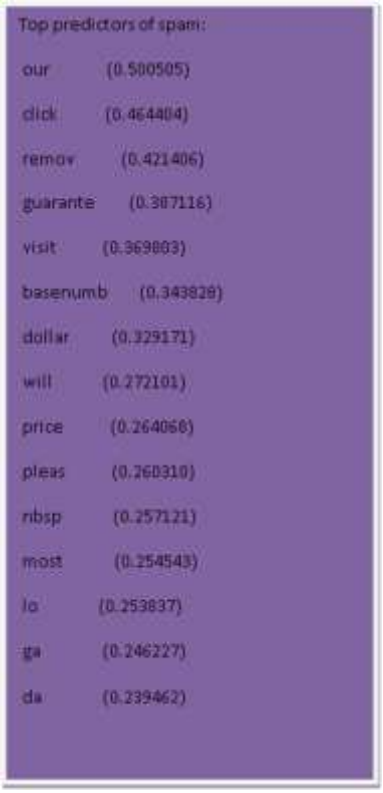

Fig 10: Junk Prognostication

Fig 11: Junk Mail Identification

References

Email mailboxes to increase to 1.2 billion worldwide by 2005. Technical Report DC #W25335 7.

Ming, Y. S., Li, H. D, He., X. M. (2016) .Contour completion without region segmentation. IEEE Transactions on Image Processing, 25(9), 3597–3611.

C. Yang, R. Harkreader and G. Gu. (2011). Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers, In: Recent Advances in Intrusion Detection, Springer Berlin/Heidelberg, 318-337.

S. Kumar, and S. Arumugam. (2015). Probabilistic Neural Network Based Classification of Spam Mails Using Particle Swarm Optimization Feature Selection, Middle-East Journal of Scientific Research, 23(5), 874-879.

N. P. DíAz, D.R. OrdáS, F. Riverola, and J.R. MéNdez. (2012). SDAI: An integral evaluation methodology for content-based spam filtering models, Expert Systems with Applications, 39(16), 2487-12500.

A. K Sharma, S. K Prajapat and M. Aslam. (2014). A Comparative Study Between Naive Bayes and Neural Network (MLP) Classifier for Spam Email Detection:, In: IJCA Proceedings on National Seminar on Recent Advances in Wireless Networks and Communications, Foundation of Computer Science (FCS), 12-16.

W. Ma, D. Tran and D. Sharma. (2009). A novel spam email detection system based on negative selection", In: Proc. of Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT'09, Seoul, Korea, 87-992.

T.S. Guzella, and W.M. Caminhas. (2009) A review of machine learning approaches to spam filtering, Expert Systems with Applications", 36(7), 10206-10222, 2009.

M. Mohamad, and A. Selamat, (2015). An evaluation on the efficiency of hybrid feature selection in spam email classification, In: Proc. of 2015 International Conference on Computer, Communications, and Control Technology (I4CT), Kuching, Sarawak, Malaysia, 27-231.

T.S. Guzella, and W.M. Caminhas, (2009), A review of machine learning approaches to Spam filtering", Expert System.

M. Mohamad, and A. Selamat, (20150. An evaluation on the efficiency of hybrid feature selection in spam email classification, In: Proc. of 2015 International Conference on Computer, Communications, and Control Technology (I4CT), Kuching, Sarawak, Malaysia, 227-231.

A. Harisinghaney, A. Dixit, S. Gupta, and A. Arora, (2014).Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm, In: Proc. of 2014 International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), Faridabad, Haryana, 153-155.

S. Youn, and D. McLeod, (2007). Efficient spam email filtering using adaptive ontology, In: Proc. of Fourth International Conference on Information Technology, Las Vegas, NV, USA, 249-254.

H. Faris, and I. Aljarah, (2015). Optimizing feedforward neural networks using Krill Herd algorithm for e-mail spam detection, In: Proc. of IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Amman, Jordan, -5.

T. Guzella, and W. Caminhas, (2000). A review of machine learning approaches to spam filtering", Exp System Application, 36(7), 10206–10222.

J.Batra, K. Bhatia, R. Sharma,and S. Bhadola, (2021). An Overview on Machine Learning Based Spam Mail Identification Approaches, International Journal of Innovative Research in Computer and Communication Engineering, vol. 9, number 7, 8987-8994.