IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 13, 2021, accepted September 22, 2021, date of publication September 29, 2021, date of current version October 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3116128

An Unsupervised Approach for Content-Based Clustering of Emails Into Spam and Ham Through Multiangular Feature Formulation

ASIF KARIM, (Member, IEEE), SAMI AZAM, (Member, IEEE), BHARANIDHARAN SHANMUGAM, AND KRISHNAN KANNOORPATTI
College of Engineering, IT and Environment, Charles Darwin University, Darwin, NT 0810, Australia
Corresponding author: Sami Azam (sami.azam@cdu.edu.au)

ABSTRACT The rapid growth of spam email attacks and the inherent malicious dynamism within those attacks on a range of social, personal and business activities warrants an intelligent and automated anti-spam framework. Attempts like malware propagation, identity theft, sensitive data pilfering, monetary as well as reputational damage are sharply increasing, endangering the privacy of the victim. Current solutions that are rather incomplete when the multidimensional feature range of email, is taken into account. We believe a methodology based on Artificial Intelligence, especially unsupervised machine learning is the way forward. This research attempts to investigating the application of unsupervised learning for the clustering of Spam and Ham emails. The overall goal of the research is to develop an unsupervised framework that solely depends on unsupervised methodologies through a clustering approach that includes multiple algorithms, primarily using the email content (body) and the subject header. The clustering has been done on a novel binary dataset of 22,000 entries of ham and spam emails, composed of ten features (reduced from eleven to ten after the feature reduction). Seven out of these ten features are unique to this study, engineered to represent impactful analytical email characteristics from a multiangular point of view. Out of five different clustering algorithms investigated in this work, OPTICS produced the optimum clustering demonstrating a 0.26% higher average efficacy than its nearest performer DBSCAN. The average balanced accuracy for OPTICS and DBSCAN was found to be ≈75.76%.

INDEX TERMS Machine learning, unsupervised learning, clustering, spam detection, spam email, spam filtering.

I. INTRODUCTION

Email is an important medium of digital communication throughout the world. Billions of individuals rely on email communication for their personal, social and business needs. Unfortunately, the ubiquity of email has made it a perfect target for scammers to turn this seemingly simple but effective communication tool into a manipulative carrier of potentially damaging outcomes. Email spamming is generally defined as the act of dispersing messages that are unsolicited, oftentimes in large volumes, using the medium of email. On the other hand, emails that are communicated for genuine, lawful and authorised purposes, are defined as Ham [1]. Spamming is deployed for both marketing purposes, and to inflict reputational and financial damage, both on the personal and the institutional front. Financial gain is regarded as the driving motivation behind spamming, generating a yearly gain for the spammers of around USD 3.5 million [2]. By the end of 2020, over 4.1 billion email accounts were registered globally [3]. Approximately 306 billion emails, in 2020 alone, were exchanged of which a lion's share of 55% were identified as spam emails [3]. These large volumes of unwanted email, apart from causing damages, waste users' time and patience as well as communication bandwidth, server memory and CPU cycles. Business Email Compromise (BEC) attacks resulted in a financial damage of around USD 3.5 billion in 2019, as reported by FBI [4]. Australian consumers and businesses, by the end of 2019, had lost over AUD 28 million due to email fraud [7]. An average person spends around 28% of a regular workweek interacting with

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

emails [5]. 38% of these emails, equivalent to ≈11% of a workweek on an average are actually relevant.

Traditionally the United States has been the most prolific source of spam emails (both phishing and marketing spam). However, the trend is on the rise in other parts of the world and in the first half of 2020, Russia has been to be the largest source [6], with over 20% of spamming worldwide [6]. The United States and Germany are closely tied in second and third position respectively with 9.64% and 9.41% of global spamming [6].

Though a number of propositions are available for spam filtration systems based on supervised or semi-supervised algorithms, work based upon unsupervised methods is virtually non-existent. A few examples of the use of K-means clustering are available but the processing overhead and general lack of practicality of these systems are bottlenecks for widespread implementation. Unsupervised algorithms have fundamentally distinct advantages over supervised methods, which we believe are key for true AI based systems. These differences will be discussed more broadly in the next section.

The contributions of this study are, **I)** Developing a novel and comprehensive database of email content and subject based features from multiple publicly available email sources. This database may be used for other relevant research, **II)** Introducing a novel feature-set, formulated based upon a characteristics of content and subject of an email, and **III)** Critically investigating the clustering outcomes of a number of unsupervised algorithms on this dataset comprising of mostly novel features representing both ham and spam emails.

II. BACKGROUND OF THE RESEARCH

An extensive number of related research attempts, both supervised and semi-supervised, have already been completed. In spite of this, until now, research initiatives that fully rely on unsupervised methodologies to separate spam emails from the legitimate ones are hard to find. This research is a comprehensive initiative at addressing this gap. In this paper, we target the subject and content of the email, while in our previous work we addressed the header and domain information. The problem domain of spamming is not confined to one particular aspect of today's email subsystem. All sub-parts need investigation.

The edge that unsupervised learning has over supervised learning, as well as the lack of research available on this topic of this study, have motivated this research. This is not to say that unsupervised methods are better than supervised or vice versa. The working procedure of unsupervised algorithm, however, even though further development is needed, is something we believe has more potential in developing highly autonomous systems leading towards a true AI based framework. Supervised algorithms need labelled data to work with, where the possible output for the corresponding input is already stated and the algorithm learns from the mapping; sourcing and managing such labelled data. This is often quite a difficult and complicated task [8]. Unsupervised clustering, on the other hand, has the advantage that it operates on unlabeled data; and requires no training. Based on the dataset, the algorithms attempt to find the set of common features within a batch of assorted items and rearrange the data points in clusters, based on the commonality [9]. In Supervised Learning, inputs demonstrating little variation in the training dataset can produce outcomes with high error rates in the inference phase, due to the fact that the model could not be trained to appropriately recognise unexpected and rare patterns. Unsupervised algorithm are often better at finding patterns or relationships among features that are too complex to detect through ordinary data analytics or observation.

This study dissects the subject header and content of an email through such unsupervised clustering and investigates the clustering performance. We also focus on developing versatile and relevant feature set from multiple angles that impact on the clustering process and could generate a unique fingerprint for spam emails. This will help us in getting an objective understanding and quantifiable knowledge about the effect of clustering. We use some common clustering algorithms with custom feature engineering based upon diversified characteristics of email content and subject. This study does not rely only on raw K-means clustering of words - the most common form of content clustering that has already been used in a number of earlier research.

III. STRUCTURE OF THE PAPER

Analysis of necessary background studies is discussed in Section 4. The section after that, Section 5, describes the proposed method briefly. Section 6 has a brief discussion on the datasets and features used in this research. Section 7 details the techniques used for the overall proposed framework, feature construction, feature selection, dataset construction, clustering and its evaluation. Section 8 describes the outcomes of the research and a few limitations that we have observed. This is followed by conclusion and some directions for future work.

IV. RELEVENT STUDIES

Though it is quite difficult find closely related work, as stated earlier, this section analyses somewhat related Machine Learning based research initiatives. Most of these are mainly unsupervised in nature or have at least deployed unsupervised learning techniques to address key aspects of the proposed system. We are mainly focusing on systems that have critically analysed the email content for their automated approach.

Basavaraju and Prabhakar [10] introduced text clustering using 'Vector Space Model (VSM)', an algebraic model for the representation of text documents as vectors of identifiers [11]. The method performed reasonably well on spam email identification. Data were represented using VSM (often known as Term Vector Model) and dimensionality reduction was carried out through a custom developed clustering framework based upon BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) and K-means algorithms.

The system demonstrated an average accuracy of just over 74% on a marginal dataset of 400 records with four different combinations of BIRCH and KNN. The authors employed raw words from the documents to formulate the VSM. A general limitation is that when spammers decide to employ character variations, for instance representing the word 'insurance' as i∗n$u∗rènce, their proposed framework will not be able to detect these variations.

Laorden *et al.* discussed a system [12] that identifies spam emails through a content based anomaly detection framework. The proposed system works by drawing a comparison between features such 'Word Frequency' to those of a ham (legitimate emails) dataset. For a significant deviation from the normal scale, a spam alert is raised. The researchers have used a variation of K-means Clustering called 'Quality Threshold (QT)', which provides an edge in downsizing the tally of vectors within the dataset that is designated as the 'normality'. Processing overhead is reduced significantly with this technique. On the other hand, the system may not work as expected for the application of language features such as Hyponyms [13], Synonyms and Metonymy. The authors have demonstrated a weighted accuracy of 92.27%. The experiment used the LingSpam dataset [61].

K-means, along with Expectation Maximization (EM) were also used by Halder *et al.* [14] to develop a framework that works on particular schemas such as stylistic characteristics or features of emails (total number of contractions and punctuations, total count of email IDs used within the body etc.). Different semantic features, such as the statistical measures of different words used in a batch of emails were also investigated. A combination of these two approaches was considered as well. The eventual cluster analysis was carried out on a dataset of 2,600 spam emails. The authors demonstrated that the method could be used to detect the composition styles of spam campaigns. Furthermore, the extracted patterns can also be used to build prototypes for prospective future identification of spam emails. When a combined approach is taken, K-means produced an 80% success rate. On the other hand, while dealing with only semantic features, EM projected a success rate of 84.6%, whereas the detection rate drops to 57.4% if a combined approach is considered. The result of the experiment was reported in terms of the 'purity' of clusters, a measure of cluster quality. The authors' area of focus [14], is generally rather limited as there is a range of critical features in spam email identification such as URLs composition, email subject headers, attachments, detailed domain as well as header information etc. which have not been included in the framework.

Unsupervised Self-Organized Map based systems have also been explored by researchers such as Cabrera-León *et al.* [15]. Their introduced system works on 13 different categories for emails. The authors started with a 4-stage preprocessing of emails (both ham and spam). In the first stage, batch-extraction of all the emails' subject and content was carried out and alphanumeric characters replaced the whitespaces. The next stage removed all the stop words and derived raw term frequency scores in addition to some other critical metadata (spam\ham) used in the processing. The third stage developed a 13-dimensional integer array to store the themes and categorize the processed texts. The preprocessing phase ends by attaching 'weights' to the words of all of the 13 categories. SOM was then deployed to build the model with 'Batch' learning method. Eventually, a threshold value was put in place to label the clusters. An accuracy of 94.4% was achieved by the framework. However, an issue with the system is that the performance for off-topic emails, was reduced, indicating potential room for further enhancement.

Padhiyar and Rekh [16] presented a semi-supervised model that is based upon K-nearest neighbor (KNN) and Naive Bayes (NB) algorithms. The authors demonstrated that this achieved better classification accuracy than a standalone KNN or NB based methods. However, in-depth inspection reveals that in all likelihood the work will not reach the expected performance when the availability of initially labelled documents is limited. The study addresses the issue by introducing Expectation Maximization (EM) algorithm to manage a dearth of labelled data, however this has not actually been implemented within the proposed system. The solution requires an effective feature selection and pre-processing segment.

V. PROPOSED APPROACH

As the name indicates, unsupervised learning based models work only with unlabelled data so no training phase is involved; whereas supervised techniques have the requirement of training over a large dataset often requiring costly data labelling [17].

Unsupervised algorithms most commonly attempt to discover a common pattern associated with the features being processed within the dataset [18]. The algorithm rearranges the data items in separate clusters. Unsupervised learning is less time consuming and computationally efficient [19], [20] than supervised approaches. In addition to the usual 'distance based' clustering [21], where specific distance metrics such as 'Euclidean Distance' [22] are used to calculate similarity between data items or objects, 'density based' clustering is often commonly used. As shown in Fig. 1 (a), our approach comprises of building a raw dataset of pre-processed contents and other features from a number of publicly available email collections of both ham and spam emails. This dataset is then converted to binary form. A feature selection process is applied through a mechanism best described as ''feature reduction through feature elimination'', which is explained later. The resulting dataset holds the most important features out and is ready for clustering.

In the subsequent step five clustering algorithms – K-modes, DBSCAN, OPTICS, K-means and Spectral are analysed as illustrated in Fig. 1 (b). Some of the unsupervised algorithms but not all, can be programmed to generate only two clusters. In this research only these algorithms have been investigated. Once the cluster formations have been analysed,

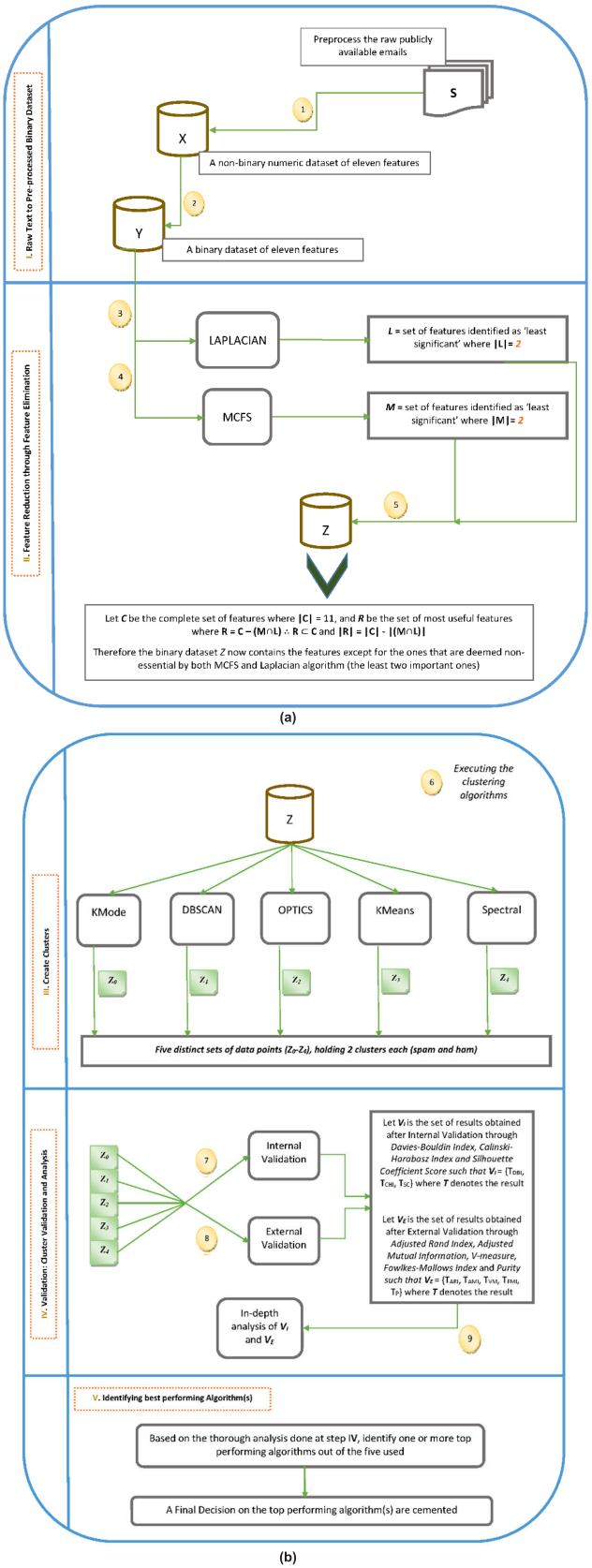


FIGURE 1. (a) The complete workflow. (b) The complete workflow.

we apply an array of external and internal validation metrics to the generated clusters to appropriately quantify the performance of the algorithms. These validation steps then indicate the top performing clustering algorithms. Section 7 provides further details of the above steps, but we will first describe the datasets and feature construction in section 6.

VI. DATASETS AND FEATURES USE

Our research is based on a custom-developed dataset where the features are of binary nature. The feature-set is a mix of some commonly used features used in similar research and some novel features (formulated based on the diverse characteristics of email content and subject). To develop the binary dataset of 22,000 records (around 68% of which are spam emails whereas the remaining are ham), where each of the records contain eleven features, we employed a number of publicly available datasets, such as the 2017 and 2018 spam collection by Guenter [23], TREC [24], the Enron Dataset [25], Hillary Clinton's political emails released by US State Department [26], highly used spam words and phrase collection by Theo Freeman [27], and Fraudulent E-mail Corpus by Tatman [28]. These datasets contain phishing spam as well as advertising spam emails and ham emails (in raw text format). The binary dataset has been made publicly available [62].

A. FEATURE CONSTRUCTION

This section highlights the features that have been used for the research. Feature f_2 and f_3 have been used previously in other studies [14], [33], [34], whereas only the concepts of f_0, f_1, f_4 and f_7 have been described in other studies but they have not been applied from the angle we have chosen in this research. Features f_5, f_6, f_8, f_9 and f_{10} are a novel set of feature that we have not found in other studies related to thus topic.

VII. DETAILED WORKFLOW

we will here describe each of the subsections as illustrated in Fig. 1 (*a* and *b*), including validation procedures and the algorithm's performance.

A. RAW TEXT TO PRE-PROCESSED BINARY DATASET

This stage includes the preprocessing of content and subject headers for each of the emails, leading first to the creation of the non-binary dataset of eleven features, and then the eventual binary dataset.

1) CONTENT AND SUBJECT HEADER PREPROCESSING

Instead of straight clustering the email contents and subjects, which is the most common way of creating clusters of probable ham and spam emails from a mixed collection, we have create novel features and included some features that have been conceptually described in other research initiatives.

TABLE 1. Feature table.

henceforth known as	*description*	*novelty*
f_0	Percentage of Spam words and phrases in email content	Conceptual use
f_1	Percentage of Spam words in Subject header	Conceptual use
f_2	Number of special characters in Subject header	Previously used
f_3	Whether the content mentions any URL with a blacklisted domain	Previously used
f_4	Percentage of alphanumeric words in the content	Conceptual use
f_5	Percentage of improper wordings in the content	Not previously used
f_6	Total number of words that are incorrect but have close match	Not previously used
f_7	Percentage of words related to currency	Conceptual use
f_8	Feature representing unique fingerprint composed of numeric hash code for the word pair distances, particularly useful for identifying spam campaigns	Not previously used
f_9	Total occurrence of words in content that had three consecutive same character	Not previously used
f_{10}	Standard Deviation derived from the distances among spam words within the content	Not previously used

Preprocessing of content and subject header is critical to feature f_0 and f_1. Textual data are highly unstructured and preprocessing allows simplification of semantically duplicate words, removes noises or words that do not have any real impact and compacts the overall text for further processing. The complete gamut of preprocessing steps allows both f_0 and f_1 to produce more accurate metrics. For the other features preprocessing has not been done. The following preprocessing techniques have been applied:

a: STRIPPING HTML TAGS

The raw text files available in public domain contain a number of HTML and other similar tags within the content and sometimes in the subject header. These tags have been removed.

b: REMOVING LONE CHARACTERS

All single characters were removed as part of the preprocessing as these did not have a meaningful impact.

c: EXPANDING THE CONTRACTIONS

Contractions are abridged version of syllables or words. These contractions were expanded to its full form; for instance, *you've* becomes *you have*.

d: STRIPPING SPECIAL CHARACTERS AND LOWERING THE CASE

Special characters - ‘‘()[]{}+-_*/, \ ∼ |%‘‘’’ were stripped off and replaced with whitespace. The remaining content and subject were fully lowercased.

e: TOKENIZATION AND REMOVAL OF STOPWORDS

Tokenization is the mechanism of breaking a document down into its individual parts called *tokens*, such as words, punctuation marks and numbers. The complete content and subject headers for individual emails are tokenized and certain *Stopwords* were removed in this stage. Stopwords are mostly pronouns, prepositions and linking verbs (such as ‘is, are’, ‘was’ etc.). We have used the python library spaCy [29] for the preprocessing, which has a rather long list of Stopwords. We have removed some words from the default spaCy Stopwords repository which we deem important for this particular research. ‘Appendix C’ contains those words.

f: LEMMATIZATION

Lemmatisation is the algorithmic process of grouping together the inflected forms of a word in order to analyze them as a single item, identified by the lemma of the word [30]. Generally many words appear in several inflected forms. For instance, the verb ’to sleep’ may appear as ’sleep’, ’sleeps’, ’slept’ or ’sleeping’. The base form, ’sleep’, is termed as the lemma for all the other versions of usage. spaCy has also been used for Lemmatisation. The Lemmatisation process had been run three times back to back to refine the preprocessed content and subject. Figure 2 shows an example of this process.

g: REMOVAL OF NON-ENGLISH WORDS

At the final stage of preprocessing, words or lemmas that are not found in English vocabulary, such as incomplete or alphanumeric tokens, are removed.

TABLE 2. Example of close match words against some invalid words.

#	*invalid word (x)*	*Close match(es) (y)*
eg_0	'a@@ain', 'attvin' or 'at$akn'	attain
eg_1	tein, tuun	turn or twin
eg_2	eztreme	extreme

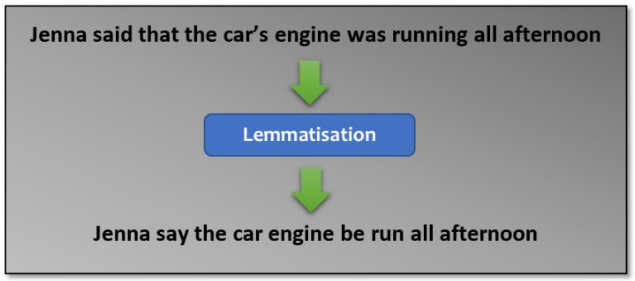


FIGURE 2. An example of Lemmatisation.

2) FEATURE CONSTRUCTION

The features which have been used to initially populate the dataset (***X*** in Fig. 1 (a)) are described below.

$\boldsymbol{F_0}$: The total number of spam words present in the content has been used in some studies. However, we have also accounted for phrases (such as 'billion dollars') instead of words only. We calculate the percentage of the total count of such spam words and phrases to that of the total words present in the content. This collection of highly probable spam words and phrases has been taken from a publicly available source [23]. It is also supported by a number of other web sources. An excerpt has been added in 'Appendix A'.

$\boldsymbol{F_1}$: Similar to $\boldsymbol{f_0}$ but applying to subject header. Only words no phrases are used for this feature.

$\boldsymbol{F_2}$: Total number of Special Characters (!#<>{@:}%[],; +-*/=_. ~''?(\)) present in the subject header.

$\boldsymbol{F_3}$: Whether the content has an URL with blacklisted domain. We have used a number of major DNSBL servers for the identification of such malicious domains.

$\boldsymbol{F_4}$: Excess number of alphanumeric strings in the body are a major indicator of spam content. This feature calculates the proportion in percentage of such occurrences.

$\boldsymbol{F_5}$: Percentage of words that are not found in English language vocabulary. For instance, ''cloze his bank account'', the feature would yield a value of 0.25 as the word ''cloze'' is not an English word. NLTK Wordlist [31] has been used in this research as a comprehensive dictionary of English language words.

$\boldsymbol{F_6}$: This feature represents the total number of words which are not in the English dictionary but have at least one valid word in the dictionary that is a close match. If ***y*** is to be considered as a *Close Match* of an invalid word ***x***, the following set of conditions must be True:

- ***a.*** The length of ***x*** has to be greater than two characters and must not contain any of these special characters - ''()[]{}+-_*/, \ ~ |%'''' to begin the process
- ***b.*** The first and last character of ***x*** and ***y*** need to be the same
- ***c.*** A variation or difference in at most two characters is allowed keeping the first and last character of both ***x*** and ***y*** same
- ***d.*** ***y*** must be of the same length as ***x***
- ***e.*** These three special characters – '@', '!' and '$' are allowed within ***x*** but cannot be the first or last character

Table 2 provides some examples on how this feature may work against invalid words; whereas Table 3 details how the examples in Table 2 satisfy the conditions (*a* to *e*).

$\boldsymbol{F_7}$: This feature provides the percentage of words related to currency, such as USD, AUD, U$ etc. The complete list of the most commonly used words that have been taken into consideration for this feature can be found in 'Appendix A'.

$\boldsymbol{F_8}$: This feature contains a hash value generated from a numeric string. The numeric string is formulated by positioning side by side the distances among high-probability spam words and phrases within the content. There are several steps before the final value is reached as outlined below:

- From the pre-processed content, spam words and phrases are first identified. Phrases are turned into single word through the usage of '_' within words. Pre-processing stages will be discussed in a later section.
- Spam words and phrases are then numbered as shown in Fig. 3. Numbers are added in a descending order primarily because of the ease in algorithm development.
- These numbered words and phrases are sorted alphabetically as shown in Table 4
- Forty-five word pairs are then derived in a view to calculate the distance between the high-probable spam words and phrases (Fig. 4). The combinations are derived using (1) where ***n*** is the aggregated total of words, ten in this case.

$$\frac{n!}{2!\,(n-2)!} \tag{1}$$

- The distance between the words in a pair is determined. For the purpose of simplicity, three word pairs out of the forty-five have been used for visually demonstrating how the distance is calculated, in Fig. 5.

TABLE 3. Breakdown of the Table 2 examples.

#	*Conditions (a to e)*	*comment*
eg_0	***a.1.*** Length (x) = 5	should be > 2
	a.2. x_contains_any_of (“()[]{}+-_*/,\~!%``”) = False	should be False
	b.1. First_character (x) = ‘a’ AND First_character (y) = ‘a’	must be same
	b.2. Last_character (x) = ‘n’ AND Last_character (y) = ‘n’	must be same
	c. a__ain => 2-characer variation, att_in => 1-characer variation, at_a_n => 2-characer variation	‘_’ represents the character that varies from (y), ‘attain’ in this case
	d. Length (x) = 5 AND Length (y) = 5	should be equal
	e. x_contains_any_of (“@!$”) = False (‘attvin’) x_contains_any_of (“@!$”) = True (‘a@@ain’, ‘at$akn’) ***e.1.*** First_or_Last_character (x) = ‘a’ *or* ‘n’	False => N/A if True then ***e.1.*** should not return any of “@!$”
eg_1	***a.1.*** Length (x) = 4	should be > 2
	a.2. x_contains_any_of (“()[]{}+-_*/,\~!%``”) = False	should be False
	b.1. First_character (x) = ‘t’ AND First_character (y) = ‘t’	must be same
	b.2. Last_character (x) = ‘n’ AND Last_character (y) = ‘n’	must be same
	c. *t__n* => 2-characer variation	‘_’ represents the character that varies from (y), ‘turn\twin’ in this case
	d. Length (x) = 4 AND Length (y) = 4	should be equal
	e. x_contains_any_of (“@!$”) = False	False => N/A
eg_2	***a.1.*** Length (x) = 7	should be > 2
	a.2. x_contains_any_of (“()[]{}+-_*/,\~!%``”) = False	should be False
	b.1. First_character (x) = ‘e’ AND First_character (y) = ‘e’	must be same
	b.2. Last_character (x) = ‘e’ AND Last_character (y) = ‘e’	must be same
	c. e_treme => 1-characer variation	‘_’ represents the character that varies from (y), ‘extreme’ in this case
	d. Length (x) = 7 AND Length (y) = 7	should be equal
	e. x_contains_any_of (“@!$”) = False	False => N/A

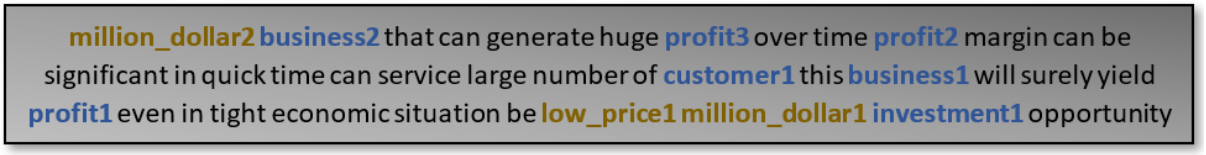


FIGURE 3. The numbering of probable spam words and phrases.

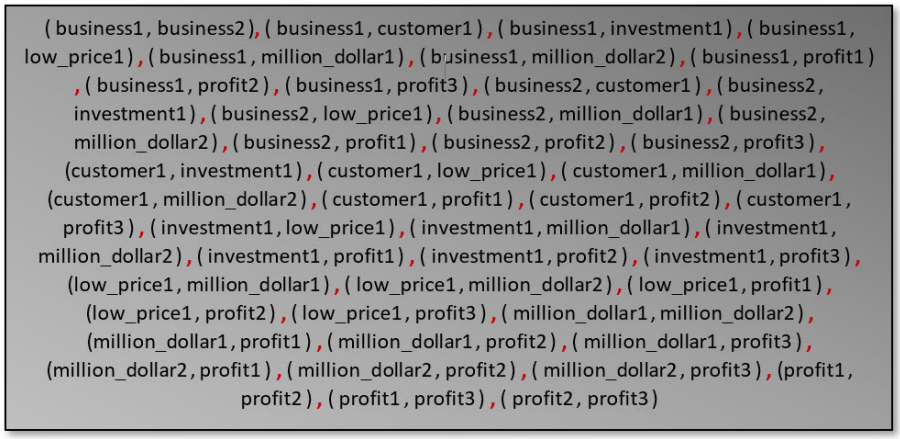


FIGURE 4. Forty-five word pairs have been obtained from ten high-probable spam words and phrases.

TABLE 4. Alphabetically sorted numbered words and phrases.

#	*alphabetically sorted*
1	business1
2	business2
3	customer1
4	investment1
5	low_price1
6	million_dollar1
7	million_dollar2
8	profit1
9	profit2
10	profit3

- All the forty-five distances are placed side by side, starting from the first word pair and sequentially moving to the last, and a space character is inserted after every fourth value. For instance, if we have six distance values such as: 22, 2, 0, 6, 10 and 35, we would construct complete strings, s, as shown in Fig. 6. The space character is added to simplify the hashing operation.
- In this step, the numeric string s is being used as the input for the hash function. The hashing algorithm that we have used here is MinHash [32].

The hash code that is generated is particularly useful for identifying spam campaigns as in the case of such campaigns, a large number of emails are spread which may have a slightly different segment of texts in the form of receiver's name, address, office and account information etc. but the majority of the content remains the same.

MinHash has a distinct advantage in such cases as slight changes in the original content does not change the hash code significantly. It will normally generate closely similar hash codes for emails having reasonably similar content and a clustering effort can produce useful results. Though we have used this feature in a different manner in this research, we do have plans to extend this idea to carry out clustering operations on spam campaigns.

The hexadecimal hash string *(Fig. 7(a))* that is now generated by MinHash is transformed to fully numeric by removing the alphabetical characters from the string *(Fig. 7(b))*, the length is also limited by stripping the first twenty-two digits as these are mostly '0's and rounding up after the eighteenth digit *(Fig. 7(c))*. Finally the number is finally changed into fraction by moving all the digits to the right of the decimal point. Figure 7 shows a hash generated by the algorithm and the subsequent steps till the resulting final fractional value.

$\boldsymbol{F_9}$: Keeps a total count of those words within the content that have three consecutive identical characters, for instance, "profiiit". With such techniques, spammers often try to evade conventional spam filtering frameworks that rely on the correct identification of suspicious words.

$\boldsymbol{F_{10}}$: This feature provides a measure of how the spam words and phrases are spread over the content of both

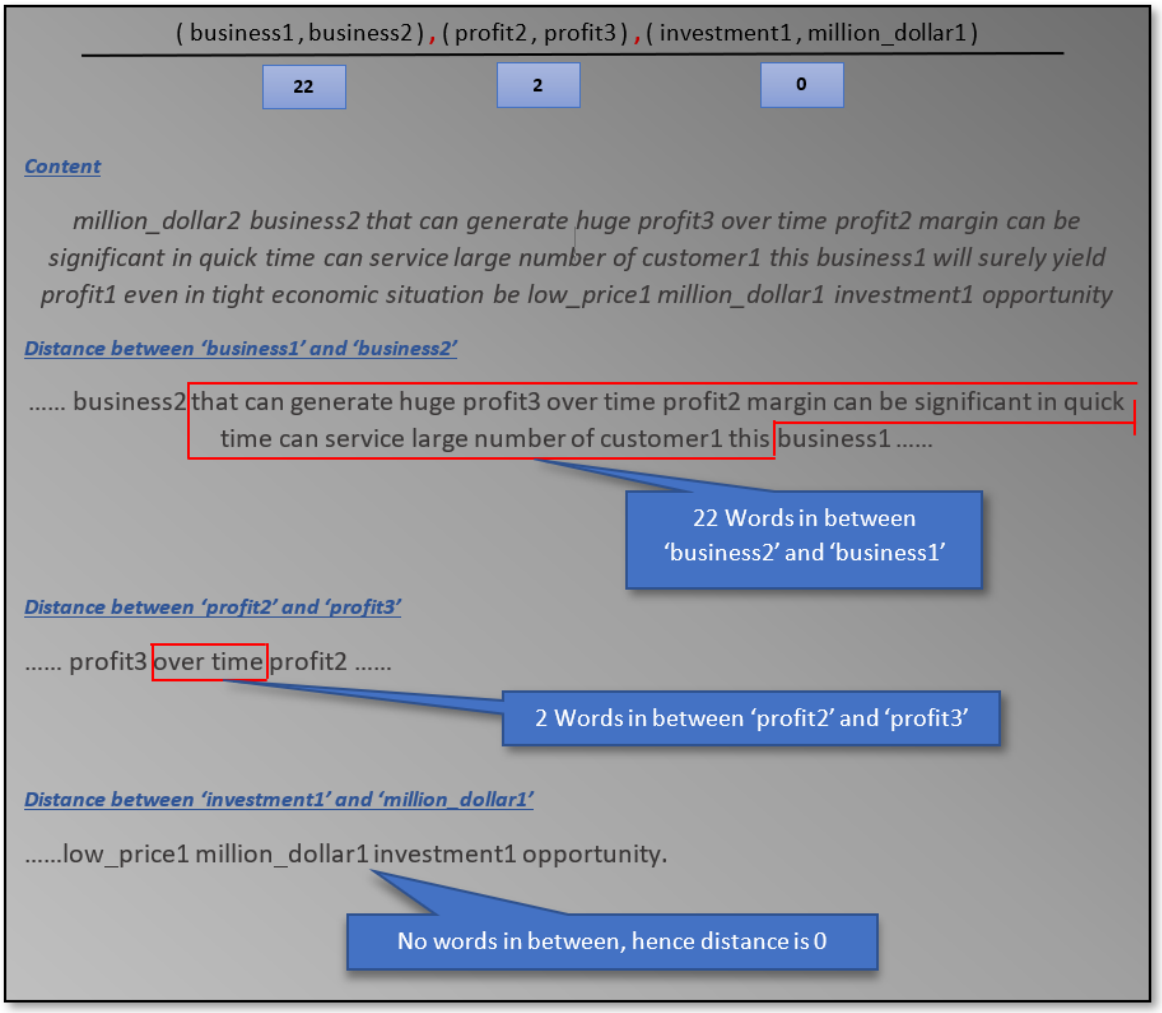


FIGURE 5. Calculating the distance between words within a word pair.

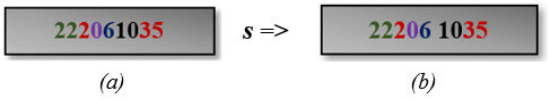


FIGURE 6. Distance string composed of six distance values (a), space character added (b).

ham and spam emails. The standard deviation, the average of squared distance from the mean, has been used on the distance points, calculated as shown in the discussion section for feature f_8. The series of resulting values produce distinct patterns of variability for both types of emails which can eventually be used as a feature of useful impact.

3) FORMULATION OF THE BINARY DATASET

Both the non-binary and the binary dataset of 22,000 entries contain eleven features. The dataset of both ham and spam email is ordered in a random fashion. To convert the dataset from non-binary (X in Fig. 1 (a)) to binary (Y in Fig. 1 (a)), we have deployed the *Standard Deviation* (*SD*) of nine of the features to be the decider; while the remaining feature (f_3) is already in binary form. For f_7, a default value of '0\1' has been used based on the extent of presence of multiple currency related tokens. In our initial exploration of the non-binary dataset, it could be observed that that the majority of features have higher numeric values for spam emails than that for ham, whereas lower scale numbers are more evenly distributed between the two classes of email. We consider the variability as impactful. The SD in this case provides a reasonable barrier of separation which is required for clustering algorithms s.

If ρ is a data point within a feature f and α is the SD of f, then its binarized equivalent f_B holds either 0 or 1 as per Table 5. Binarization often reduces the randomness in a dataset, increasing the processing efficiency. An excerpt of the dataset is shown in Fig. 8.

B. FEATURE REDUCTION THROUGH FEATURE ELIMINATION

We used both Laplacian [35] and Multi-Cluster based Feature Selection (MCFS) [36], two unsupervised feature

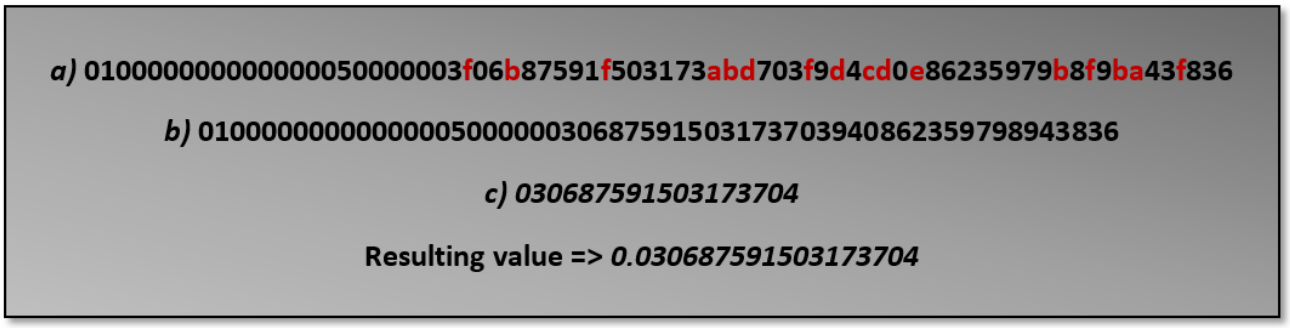


FIGURE 7. An example of a hash and the corresponding converted value.

TABLE 5. Conditions for feature binarization.

features	*condition*	*note*
$f_0, f_1, f_4, f_5, f_6, f_8, f_9, f_{10}$	$f_B = \begin{cases} 0, & if\ \rho > \alpha \\ 1, & otherwise \end{cases}$	
f_3	*as is*	
f_2	$f_B = \begin{cases} 1, & if\ \rho > \alpha \\ 0, & otherwise \end{cases}$	*reverse-binariztion yielded better outcome*
f_7	$f_B = \begin{cases} 0, & if\ \rho > 1 \\ 1, & otherwise \end{cases}$	*standard deviation has not been applied here, instead total count of currency tokens used*

sub_spm_perc	con_spm_perc	sub_tot_sp_char	con_bl_url	con_alpha	con_imw	con_clm	con_cur	con_spm_dist_hash_short	con_spm_dist_sd
1	0	1	0	1	1	0	1	0	0
1	0	1	0	1	1	1	0	0	0
1	0	0	0	1	1	0	0	1	0
1	0	0	0	0	1	1	0	0	0
1	0	1	0	0	1	0	0	0	0
0	1	1	1	1	1	1	0	0	0
1	0	0	1	1	1	0	1	0	0
1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	1	0	0
0	1	1	0	0	1	0	0	1	0
1	1	1	0	1	1	1	1	0	0
1	0	1	0	1	1	0	1	1	0
1	1	1	0	0	1	0	1	0	0
1	0	1	0	1	1	0	1	0	0
0	0	1	0	1	1	0	1	0	1
0	0	1	0	1	1	1	1	1	0
0	1	1	0	1	1	0	1	0	1

FIGURE 8. An excerpt of the dataset used.

selection algorithms, on the binary dataset. There were a few more choices of feature selection algorithms but these were not properly scalable to fairly large datasets. Before shedding light on our proposed feature reduction method, a brief discussion on the two algorithms is in order:

1) MULTI-CLUSTER-BASED FEATURE SELECTION (MCFS)

Generally MCFS produces an optimized feature-set through the application of an '$L1-$regularized least-squares' problem [36] as pointed out in (2). MCFS can preserve the multi-cluster structure of the inputted feature-set, while spectral analysis is carried out over the data points to determine

the correlations among features. The unsupervised characteristics of the algorithm allow it to work out the correlation even in the absence of corresponding labels.

$$\min_{l_k} ||q_k - X^T l_k||^2 + \beta |l_k| \tag{2}$$

In (2), $\boldsymbol{Q}$ is the 'flat' embedding for all instances where $\boldsymbol{Q} = [q_1, q_2, \ldots, q_k]$, l_k is the N-dimensional vector and $|l_k| = \sum_{i=0}^{N} |l_{ki}|$ denotes the $L1$-norm of l_k.

To rank the features, they are assigned a score (termed 'MCFS Score') based on the maximum coefficient of sparse representation. The most useful features are subsequently ranked in descending order. MCFS performs particularly well when the total features is less than fifty [37].

2) LAPLACIAN SCORE FOR FEATURE SELECTION

The key working principle for this algorithm is the inference that data points placed within the same class are often closer to each other; therefore it is possible to measure the importance or gravity of a feature through its degree or capability of locality preservation. Laplacian score initiates the procedure by embedding the data points on a nearest neighbor graph $\boldsymbol{T}$ containing $\boldsymbol{m}$ nodes. The $\boldsymbol{i}^{\text{th}}$ node stands for the element $\boldsymbol{z_i}$. The graph facilitates a connection to $\boldsymbol{z_i}$ with another node or element $\boldsymbol{z_j}$, which belongs to k nearest neighbors of $\boldsymbol{z_i}$. The Weight Matrix $\boldsymbol{W}$ of $\boldsymbol{T}$, defined using (3), illustrates the local structure of the data space [7].

$$\boldsymbol{W} = \begin{cases} e^{-\frac{||z_i - z_j||^2}{c}}, & \textit{if } z_i \in kNN\ (z_j) \textit{ or vice versa} \\ 0 & \textit{otherwise} \end{cases} \tag{3}$$

$\boldsymbol{c}$ is a suitably chosen constant, and a *graph Laplacian*, $\boldsymbol{L}$, (additional discussion in the 'Spectral Clustering' section), is constructed from $\boldsymbol{W}$. For every feature $\boldsymbol{r}$, $\boldsymbol{LS}$ - a Laplacian score, is then derived using (4), where $\boldsymbol{D}$ is the *Diagonal Matrix* of $\boldsymbol{W}$. Features are then ranked accordingly [7].

$$\boldsymbol{LS_r} = \frac{(f_r - \frac{f_r^V D1}{1^V D1} * 1)^V * L * (f_r - \frac{f_r^V D1}{1^V D1} * 1)}{(f_r - \frac{f_r^V D1}{1^V D1} * 1)^V * D * (f_r - \frac{f_r^V D1}{1^V D1} * 1)} \tag{4}$$

3) PROPOSED METHOD FOR FEATURE SELECTION

The primary aim of adopting the process of Feature Selection is to have confidence that the feature(s) that are eliminated indeed offer significantly less variation. Once the ranking is done by both the algorithms on the full feature set of eleven features, the two least important features produced by both algorithms are separated, that is the two features that were at the bottom of the ranked feature list for each of the algorithms. Out of these two sets ($\boldsymbol{s_1}$ and $\boldsymbol{s_2}$) of four features in total, $\boldsymbol{f_9}$ was the common one. It has therefore been removed from the final set of features resulting in a set of ten features for clustering purposes. Note that if no common feature is present within $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$, the original set of features will have to be used for clustering. Pseudocode 1 explains the feature reduction procedure, where $\boldsymbol{S_{af}}$ is the set of all features and $\boldsymbol{R}$ is the set of most important features.

Pseudocode 1 Pseudocode Describing the Feature Reduction Algorithm

BEGIN

1. **LET [1], {LP}, {C_S}, {R}**= { }
2. **LET** $\boldsymbol{n} = 2$
3. MCFS (S_{af})
   - ***a.*[1]** = Set of the penultimate ***2*** of the least important features
4. Laplacian_Score (S_{af})
   - ***a.*{LP}** = Set of the penultimate ***2*** of the least important features
5. {C_S} = Set holding all features (11) in S_{af}
6. **{R}** = C_S– **(MC∩LP)**

END

As can be seen from Pseudocode 1, the dataset $\boldsymbol{Z}$ of Fig. 1 (a) is composed of the features in set R. This dataset is now ready for clustering, as detailed out in the next section.

C. APPLICATION OF UNSUPERVISED CLUSTERING ALGORITHMS

This section will discuss the algorithms that have been employed for clustering purposes (Fig. 1 (b)). As mentioned previously, only those algorithms that allow us to parametrically control the number of clusters created, have been deployed. In the subsequent sub-sections, the resulting clusters will be investigated through 3D Scatterplot visualisation.

1) DBSCAN (DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE)

DBSCAN is an algorithm where 'Density based Clustering' has been implemented over centroid-based clustering as we will see in K-means. Distinctive clusters that are present within the set of data points can be recognized through density based clustering (an unsupervised learning method). The algorithm assumes that clusters in a data space usually form a contiguous zone of high point density, which can be disengaged from other similar dense regions [38] [7]. DBSCAN initiates the process by calculating an approximation of 'density' and then subsequently pushes those data points that are placed in the low dense areas further away from each other, as well as from zones of high density. The 'Mutual Reachability Distance (MRD) (5)', achieves the task [7].

$$x_{\text{mrd}-\text{d}}(u, v) = \max\{\text{core}_d(u), \text{core}_d(v), x(u, v)\} \tag{5}$$

$\text{core}_k(p)$ or the 'Core Distance' is calculated for parameter $\boldsymbol{d}$ for a point $\boldsymbol{u}$ as the minimum value of radius necessary to classify $\boldsymbol{u}$ as a core point. However in case the given point is not a Core point, then its Core Distance is undefined. Fig. 9 highlights the concept of Core Distance for $\mathbf{d} = 4$.

So, if the minimum points per cluster is 'large', then the linking 'core distance' will turn out to be larger as well. $x\ (u, v)$ is the original metric distance between $\boldsymbol{u}$ and $\boldsymbol{v}$. Now those dense points that have reasonably low core distances,

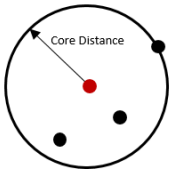


FIGURE 9. Core Distance.

are to be kept at the same distance apart from each other while the points that are sparsely placed will be pushed apart (with a distance which at least equals their core distance) to be positioned away from any other point. DBSCAN then invokes Minimum Spanning Tree to pinpoint these dense regions before the extraction of the resulting clusters [7].

2) K-MEANS

K-means is a key clustering algorithms (centroid-based) that is able to cluster together data points that are similar enough to generate some underlying pattern. The final output from K-means is obtained by an algorithmic method called *iterative refinement*. K-means minimises the sum of the squared distance between the cluster's centroid and the data points. Cluster centroid is the arithmetic mean of all the data points belonging to that cluster. **K** is observant of the number of groups (or the initial number of clusters that has to be inputted). Based on the identified similarities among the features, each data point is iteratively allocated to one of these groups of clusters [39]. In the majority of the cases K-means applies 'Euclidean Distance' to work out the distance between two data items or points ($\boldsymbol{V^n}$ and $\boldsymbol{V^m}$) as shown in (6) [7], [40].

$$Dist\left(V^n, V^m\right) = \sqrt{\sum_{i=1}^{D}(V_i^n - V_i^m)^2} \tag{6}$$

3) SPECTRAL

Spectral clustering, a graph-based clustering technique, has improved performance in some cases compared to K-means.

The clustering process first generates a 'Similarity graph' (a non-negative symmetric graph) of $\boldsymbol{M}$ objects. The most universal way of constructing this 'Similarity graph' is the ε- neighbourhood graphs or otherwise called Epsilon neighbourhood graphs. Generally, K-means is internally employed by Spectral clustering to achieve the grouping of objects into $\boldsymbol{k}$ clusters but prior to that, a feature vector is created for each of the $\boldsymbol{M}$ objects by identifying the first $\boldsymbol{k}$ eigenvectors of its Laplacian matrix [7].

Graph Laplacian matrices, $\boldsymbol{L}$, [41] are the core of Spectral algorithm, demonstrated in (7).

$$L = D - Y \tag{7}$$

where, $\boldsymbol{Y}$ is a 'Adjacency matrix' having $\mathbf{Y}_{ij} \geq 0$ of graph $\boldsymbol{R}$. $\boldsymbol{D}$ is the 'Diagonal matrix' of $\boldsymbol{Y}$. A normalised class of Laplacian matrix, L_{xy}, is often defined as in (8), where the points in the Diagonal matrix are denoted by $\boldsymbol{d}$.

$$\boldsymbol{L_{xy}}\,(\boldsymbol{R}) = \begin{cases} \mathbf{1} & (\boldsymbol{if\ i = x\ and\ q \neq 0}) \\ -\frac{1}{\sqrt{d_x d_y}} & (\boldsymbol{if\ x\ and\ y\ are\ adjacent}) \\ \mathbf{0} & (\boldsymbol{otherwise}) \end{cases} \tag{8}$$

4) K-MODES

K-modes extends K-means by the application of Modes in place of Means for clustering, in addition a rather straightforward matching dissimilarity method for categorical objects as used. K-modes also attempts to minimize the cost function associated with clustering through a frequency-based procedure in a view to updating the modes [42].

A dissimilarity measure for $\boldsymbol{A^1}$ and $\boldsymbol{A^2}$, two $\boldsymbol{n}$-dimensional vectors, can be generated using (9). The number of total mismatch is inversely proportional to the degree of similarity between $\boldsymbol{A^1}$ and $\boldsymbol{A^2}$. This dissimilarity $\boldsymbol{d}$ is (9):

$$\boldsymbol{d}\left(\boldsymbol{A^1}, \boldsymbol{A^2}\right) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{\delta}(\boldsymbol{a_j^1}, \boldsymbol{a_j^2}) \tag{9}$$

with

$$\delta(a_j^1, a_j^2) = \begin{cases} \mathbf{0},\ \mathbf{if}\ a_j^1 = a_j^2 \\ \mathbf{1},\ \mathbf{if}\ a_j^1 \neq a_j^2 \end{cases}$$

5) OPTICS (Ordering POINTS TO IDENTIFY CLUSTER STRUCTURE)

OPTICS is another useful clustering algorithms (Density-based). It is similar to DBSCAN in its functionality. OPTICS is able to efficiently handle clusters of varying sizes. The algorithm first calculates an 'Ordering' of all objects in the inputted dataset. The process is continued by identifying core samples of high density and then expanding these into the necessary number of clusters. For each of the points or objects within that dataset, an appropriate 'Reachability distance' and core-distance are saved. The reachability distance, $\boldsymbol{r\_d}$, of a point $\boldsymbol{q}$ in reference to another point $\boldsymbol{p}$ is the shortest distance from $\boldsymbol{p}$ if $\boldsymbol{p}$ is considered a core object. Core objects are the ones that have dense neighbourhoods. Generally $\boldsymbol{p}$ is considered a core point or object if, within its ε-neighbourhood $\boldsymbol{N_\epsilon}\,(\boldsymbol{p})$, at least min_pts points can be detected, including itself. $\boldsymbol{p}$ cannot also be smaller than the core distance, $\boldsymbol{c\_d}$, of q as pointed out in (10) [43]. Epsilon, ε, denotes the maximum distance that is considered, whereas min_pts indicates the minimum number of points required for the formation of a cluster [7].

$$\boldsymbol{r\_d_{\varepsilon,min\_pts}(q,p)} = \begin{cases} \boldsymbol{undefined} \\ \max(\boldsymbol{c\_d_{\varepsilon,min\_pts}(p)}, \boldsymbol{d(p,q)}) \\ \begin{cases} \boldsymbol{if}\ \ |\boldsymbol{N_\epsilon}\,(\boldsymbol{p})| < \boldsymbol{min\_pts} \\ \boldsymbol{otherwise} \end{cases} \end{cases} \tag{10}$$

OPTICS maintains a linear list that renders the density-based clustering structure of the data, and is often called OrderSeeds, to generate the resulting ordering. In this list, objects are sorted according to the reachability-distance with respect to their respective closest core points or objects.

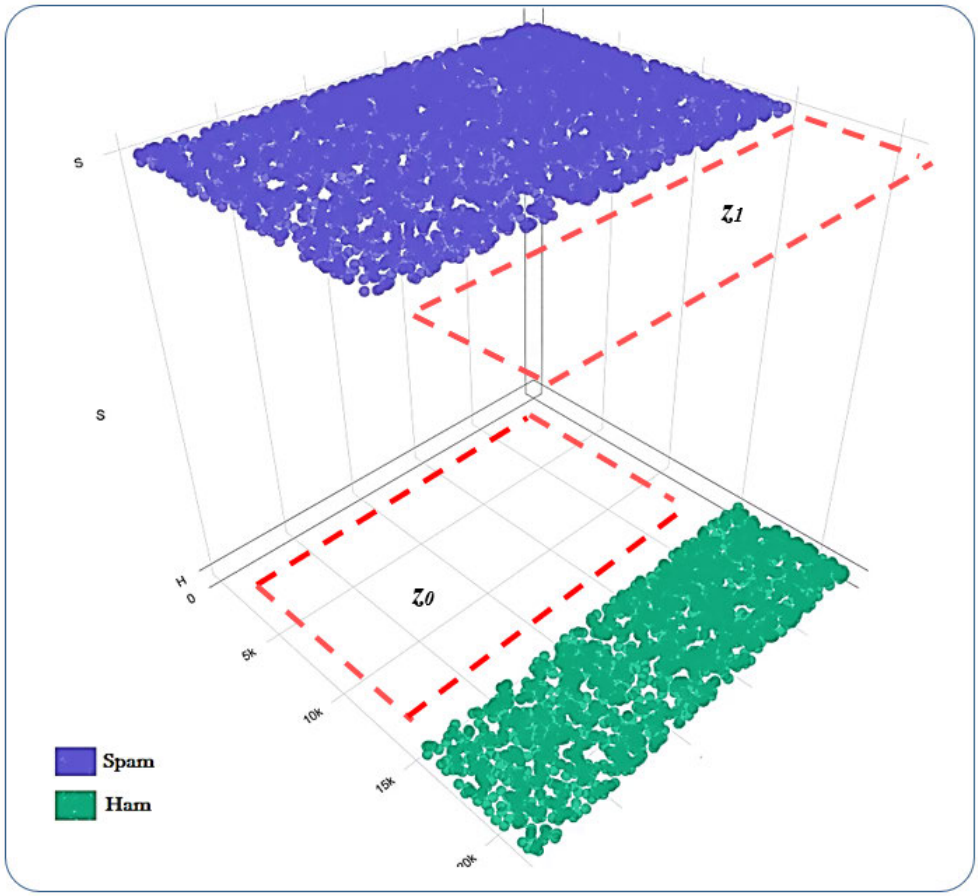


FIGURE 10. Perfect clusters in an ideal setting.

D. CLUSTERS PRODUCED BY THE ALGORITHMS

Before the application the clustering algorithms to our dataset, Hopkins Statistic [44] were applied to evaluate the probable existence of relevant, cluster-like non-random regions within the data. It will indicate whether the dataset truly contains meaningful clusters. The algorithm reported a probability of 94.44%, indicating high chance of relevant clusters being present.

In an ideal scenario, where an algorithm produces 100% correct results, it would generate clusters as shown in Fig. 10. As can be seen in Fig. 10, all the data points within the dataset constitute perfect clusters (no overlapping is observed). That is, no data points representing ham have been misclassified as spam emails. In the context of the 3D shape of the figure, we can say that 'data points are not heading upward from bottom plane to z_1 zone (marked in red dotted box)'. Also, no spam emails have been misclassified as ham or, in the context of the 3D shape of the figure, 'data points are not heading downward to the bottom plane at z_0 zone'. In reality however, it will not be as accurate or perfect as this as there will always be some degree of misclassification, and probable mixing of data points from the clusters.

A visual inspection of the algorithms' clustering patterns of our dataset is included in the next section. Algorithms that tend to get closer to the ideal clustering illustrated above, indicate better performance.

1) CLUSTERS PRODUCED BY K-MODES

Figure 11 visualizes the clustering produced by K-modes. It is evident that a high number of data points are placed into the regions of misclassification for both ham and spam emails. In fact the misclassification rate is higher for ham than that for spam emails. Thus the clustering show mixed to poor output, which is not acceptable.

2) CLUSTERS PRODUCED BY K-MEANS

The clustering structures effected by K-means can be seen in Fig. 12. The resulting clusters, as can be observed, have an extremely close similarity with the ones produced by K-modes. Therefore the performance for K-means is not at the level of reasonable satisfaction either.

3) CLUSTERS PRODUCED BY SPECTRAL

Spectral clustering, as picturised in Fig. 13, also failed to achieve noticeable improvement over K-modes and K-means.

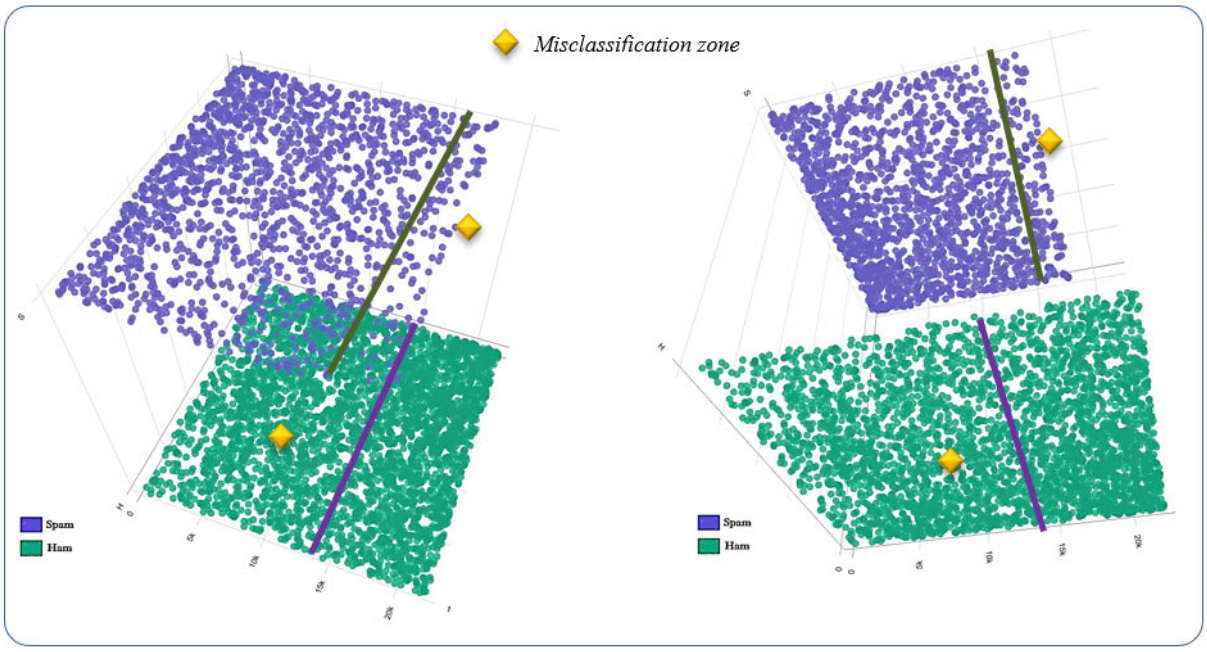


FIGURE 11. Top-down (Left) and Bottom-up (Right) views of clusters (K-MODES).

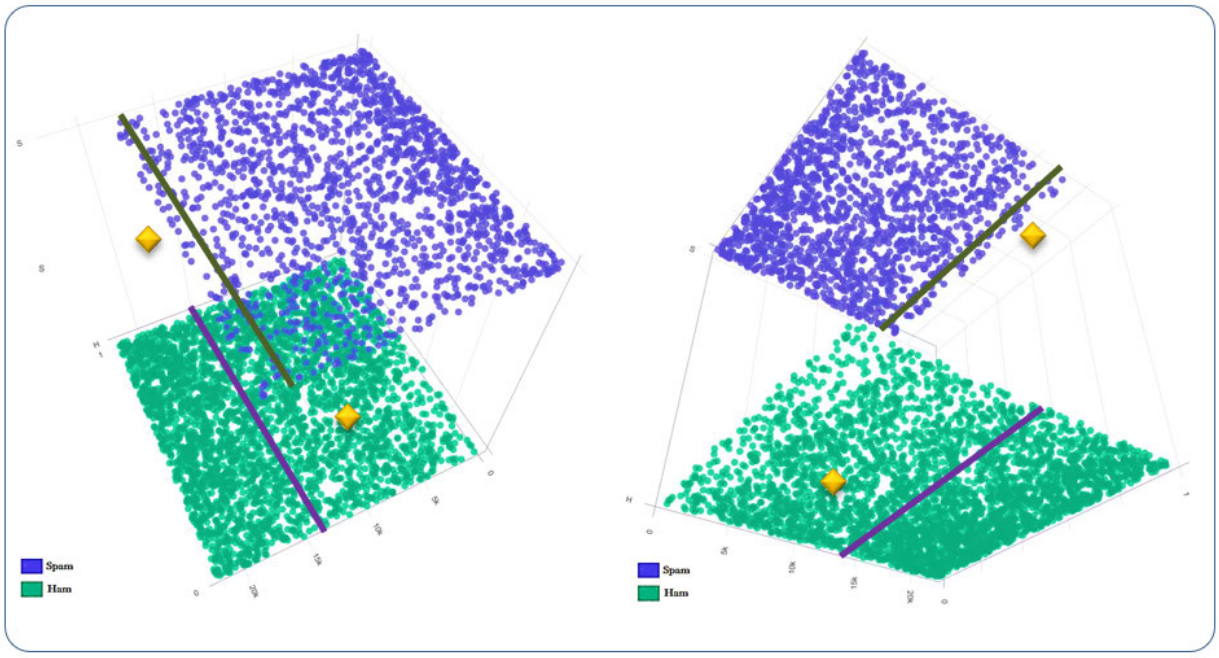


FIGURE 12. Top-down (Left) and Bottom-up (Right) views of clusters (K-MEANS).

In fact the misclassification rate is higher for both ham and spam email than for the previous two algorithms.

4) CLUSTERS PRODUCED BY OPTICS

Clustering produced by OPTICS, as depicted in Fig. 14, clearly was more accurate and robust ones than the previous three algorithms. For both ham and spam emails, the misclassification rates are considerably lower, indicating an improved quality of the clusters. The misclassification rate for spam emails has fallen sharply in this instance.

5) CLUSTERS PRODUCED BY DBSCAN

Clusters produced by DBSCAN, as portrayed in Fig. 15, are almost identical to those of OPTICS, though the

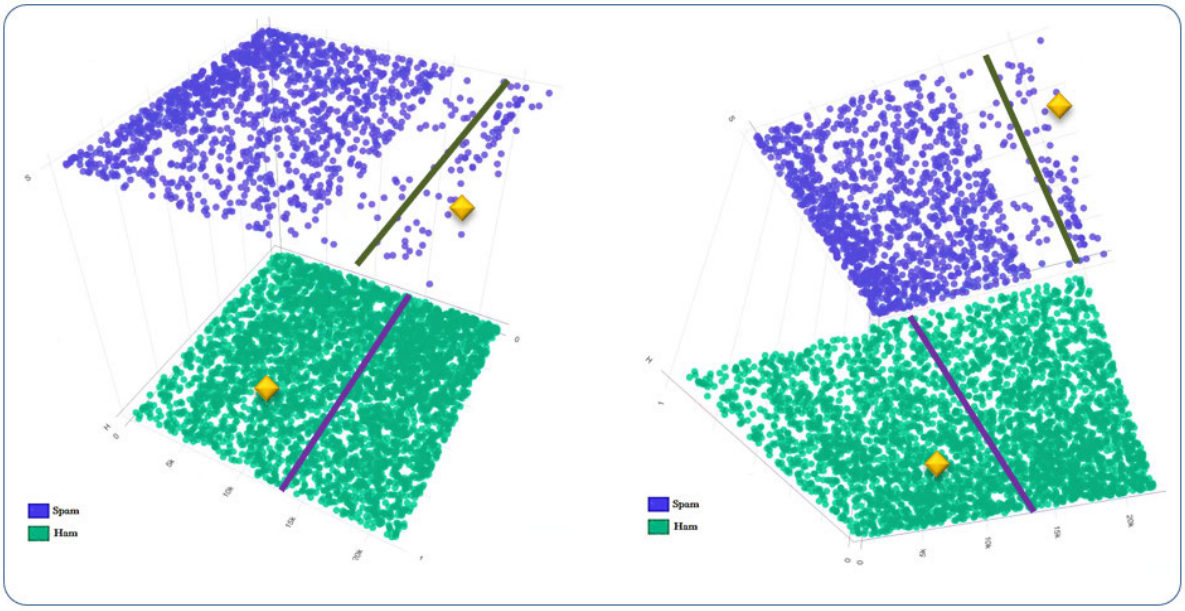


FIGURE 13. Top-down (Left) and Bottom-up (Right) views of clusters (Spectral).

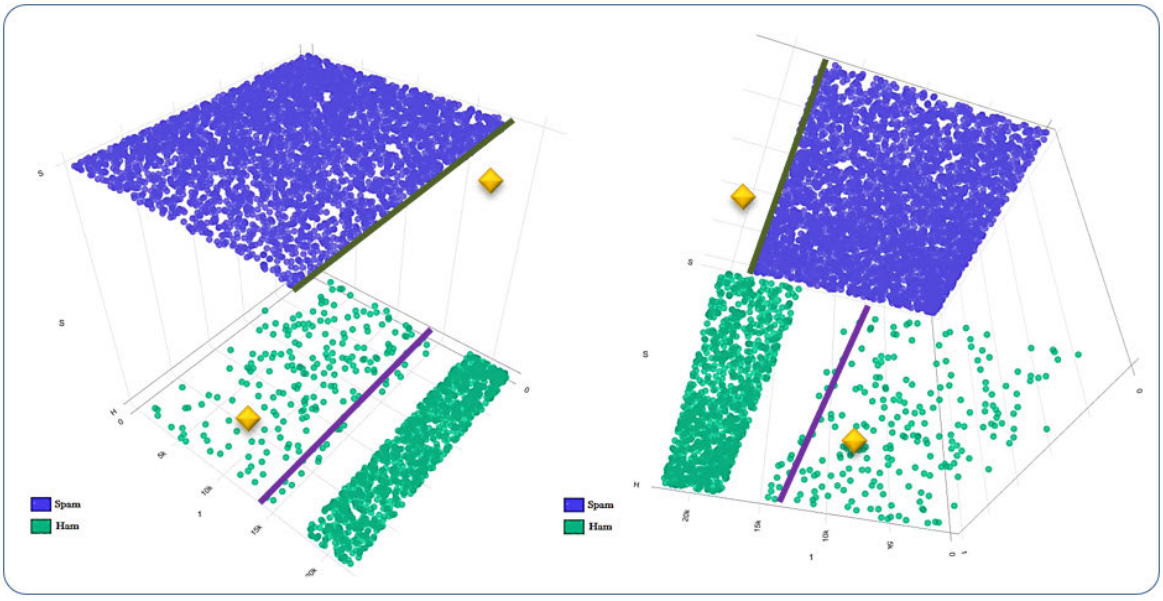


FIGURE 14. Top-down (Left) and Bottom-up (Right) views of clusters (OPTICS).

misclassification rate is slightly higher. However, the performance is considerably better than the rest of the algorithms.

The aforementioned cluster visualisation and the related discussion clarifies the fact that both OPTICS and DBSCAN performed well, while the K-modes, K-means and Spectral did not. However, to reach a quantifiable conclusion with, we will be carrying out other validation procedures in the following section.

Python's scikit-learn's [45] have been used for the implementation of both clustering and validation purposes.

E. CLUSTER VALIDATION

To have an objective knowledge into the algorithms' performance and to quantify the 'Goodness' of the resulting clusters, an array of highly relevant validation measures has been applied. We have employed both *Internal* and *External* validation methods:

- ***Internal:***

Internal validation techniques evaluate how closely data points are positioned to each other inside the same cluster,

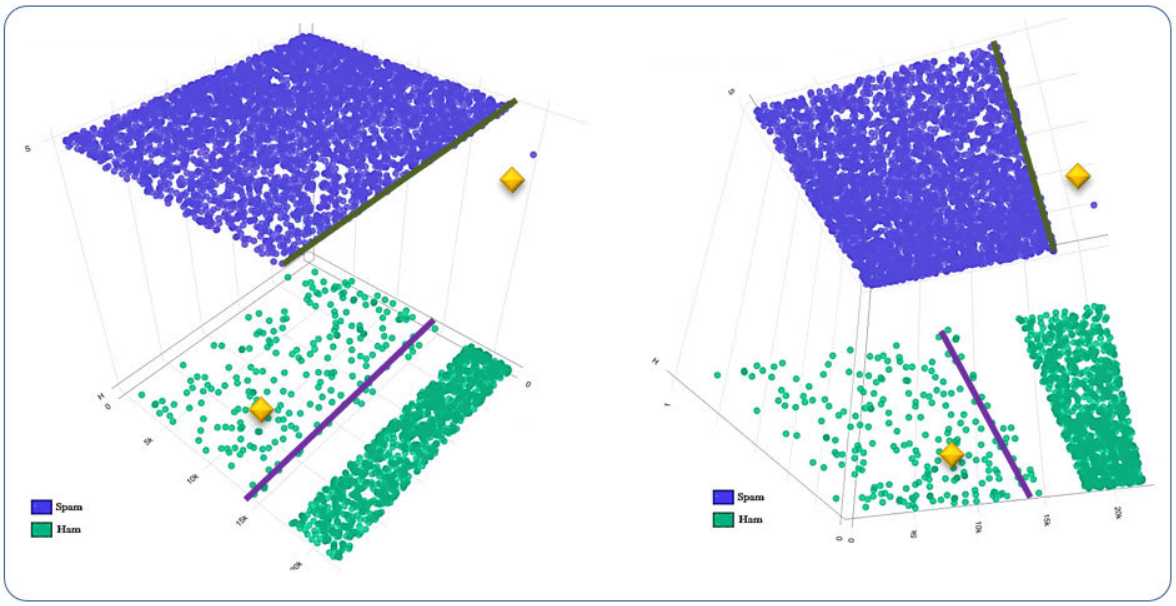


FIGURE 15. Top-down (Left) and Bottom-up (Right) views of clusters (DBSCAN).

also known as 'Compactness' [46]. In addition, how strongly a pair of data points is connected to each other within the same cluster compared to other immediate data points placed outside the cluster is also determined. This is often known as 'Connectedness'. Such validations do not require any previous ground-truths or cluster labelling. Clusters showing minimal 'Connectedness' and 'Compactness' are considered to have been well-formed.

- ***External:***

External validation techniques measure the extent to which cluster labels match the externally supplied class labels [47]. Due to the custom-built nature of our dataset, we have the option of using external measures as the class labels were available. However, except for the validation purposes outlined in this section, these class labels have not been applied in any other processes. Validation methods shown in Table 6 have been used:

1) INTERNAL VALIDATION

In this section, we will examine, using several internal metrics, how well the clusters have been formed.

a: DAVIES-BOULDIN INDEX (DBI)

The Davies-Boulding Index (DBI) metric validates the algorithms on the basis of the ratio of within-cluster distances to between-cluster distances [48]. Better clustering is indicated by smaller outcome. In this study we have employed the reverse of Davies-Bouldin Index i.e. (2− Davies-Bouldin Index, as the integer part of the largest DBI reported is '2'). As it reverses the direction of the index it provides more consistency with other indices evaluated in this work, without compromising the overall outcome. DBI can be measured for

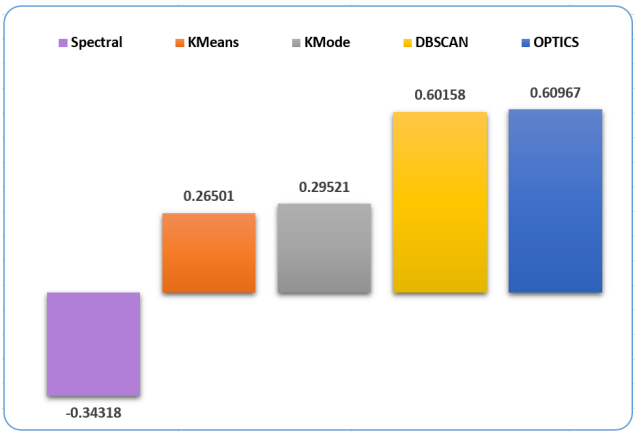


FIGURE 16. Validation results for Davies-Bouldin Index.

any value of ***n_cluster*** (n_c) using (11) [49], where d is the Euclidian Distance between the points, f_j is the cluster j where x_j is the centroid,

$$DBI = \frac{1}{n_c}\sum_{j=1}^{n_c}\max_{v=1..n_c, v\neq j}(R_{jv}) \tag{11}$$

$$\text{where, } R_{jv} = \frac{\frac{1}{||f_j||}\sum_{y\in f_j} d(y,x_j)+\frac{1}{||f_v||}\sum_{y\in f_v} d(y,x_v)}{d(x_j,x_v)}$$

From Fig. 16 it can be observed that the metric considers DBSCAN and OPTICS to be 'almost' similar or nearly equal performers while K-modes and K-means are considered as average at best, as they were far behind OPTICS and DBSCAN in terms of cluster quality, followed by Spectral which also performed poorly. This agrees with the knowledge

TABLE 6. Validation measures used in this study.

TYPE	METHOD	CRITERIA
Internal	▪ Davies-Bouldin Index ▪ Calinski-Harabasz Index ▪ Silhouette Coefficient Score	For Davies-Bouldin Index, smaller values indicate better defined clusters For Calinski-Harabasz Index and Silhouette Coefficient Score, higher score relates to a model with better defined clusters
	▪ Adjusted Rand Index ▪ Adjusted Mutual Information ▪ V-measure ▪ Fowlkes-Mallows Index ▪ Purity	Closer to 1 is optimum; ≤ 0 is poor

that we have attained so far through the Scatterplot visualisation (section 7.4).

b: CALINSKI-HARABASZ INDEX (CHI)

CHI compares the average between- and within cluster sum of squares [50] to report an evaluation on the cluster validity [50]. A higher value indicates better clustering. The index, $\boldsymbol{CH_k}$, can be defined as in (12) [51], where $\boldsymbol{X_b}$ is the between-cluster variance, and $\boldsymbol{X_w}$ stands for within-cluster variance. The total number of clusters is denoted by $\boldsymbol{k}$ and $\boldsymbol{N}$ is the total number of observations.

$$\boldsymbol{CH_k} = \frac{\boldsymbol{X_b}}{\boldsymbol{X_w}} \boldsymbol{x} \frac{\boldsymbol{N}-\boldsymbol{1}}{\boldsymbol{K}-\boldsymbol{1}} \tag{12}$$

Results returned by this index might be rather large, as it does not have any maximum upper bounds. We have confined the metric outcomes within 0 and 1 in this research to maintain conformity with other indices. This alteration may have marginally reduced the differences among various algorithms, but the original trend is preserved nonetheless.

The modification involves a minimal variation of Sigmoid function, as depicted in (13). For each output $\boldsymbol{y}$, a corresponding value $\boldsymbol{k}$ is obtained after squashing such that $\{\mathbf{k}: \mathbf{k} > \mathbf{0}$ **and** $\mathbf{k} < \mathbf{1}\}$. $\boldsymbol{n}$ is the length of the integer segment of the highest value as produced by the Index.

$$\boldsymbol{f}\,(\boldsymbol{y}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{y}/\boldsymbol{n}}} \tag{13}$$

The index (Fig. 17) indicates that the algorithms have performed almost equally, with K-means and K-modes slightly ahead of other algorithms. However, in reality, the differentiation or gap between the algorithms (especially DBSCAN and OPTICS to the rest) more substantial than what appears in the figure. However the trend does not deviate much.

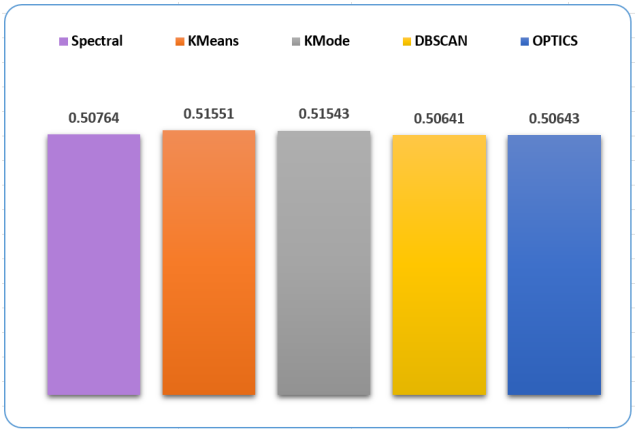


FIGURE 17. Validation results for Calinski-Harabasz Index.

c: SILHOUETTE COEFFICIENT SCORE

A widely accepted validation technique. The Silhouette Coefficient score, s, can be obtained for each of the samples using the *intra-cluster* distance (mean of within-cluster distance) $\boldsymbol{a}$ and the mean nearest-cluster distance $\boldsymbol{b}$, using (14) [52].

$$\boldsymbol{s} = \frac{(\boldsymbol{b}-\boldsymbol{p})}{\boldsymbol{max}(\boldsymbol{a},\boldsymbol{b})} \tag{14}$$

where, $\boldsymbol{b}$ denotes the distance between a sample and the closest cluster to which the sample does not belong.

The Silhouette Coefficient Score, as shown in Fig. 18, projects a rather contradictory picture to that of the scatterplots as K-means and K-modes are identified as the top performing algorithms whereas OPTICS and DBSCAN had disproportionately substandard results.

d: SUMMARISING THE INTERNAL VALIDATION OUTCOMES

Figure 19 tabulates a summarised picture of the outcomes of the internal validations, adding up the positions for each of

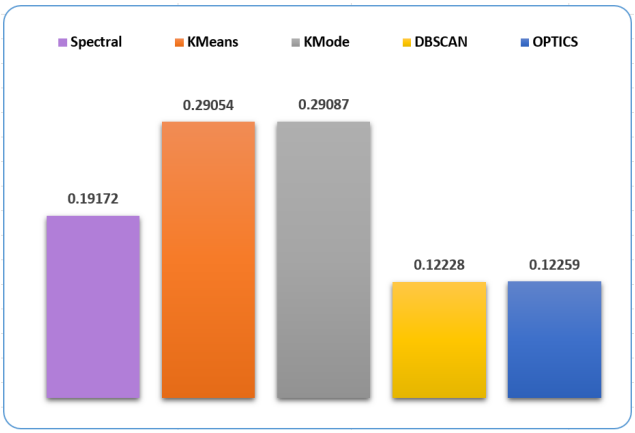


FIGURE 18. Validation results for Silhouette Coefficient Score.

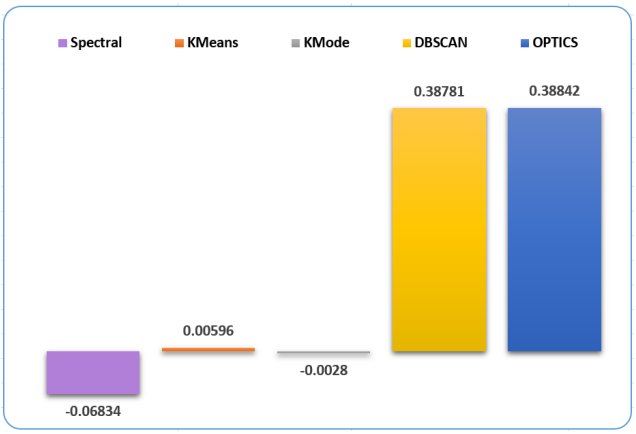


FIGURE 20. Validation results for Adjusted Rand Index.

ALGORITHM	∑ POSITION -1	∑ POSITION -2	∑ POSITION -3	∑ POSITION -4	∑ POSITION -5
Spectral	-	-	2	-	1
DBSCAN	-	1	-	1	1
OPTICS	1	-	-	1	1
K-means	1	1	-	1	-
K-mode	1	1	1	-	-

FIGURE 19. A summary of internal validation outcomes.

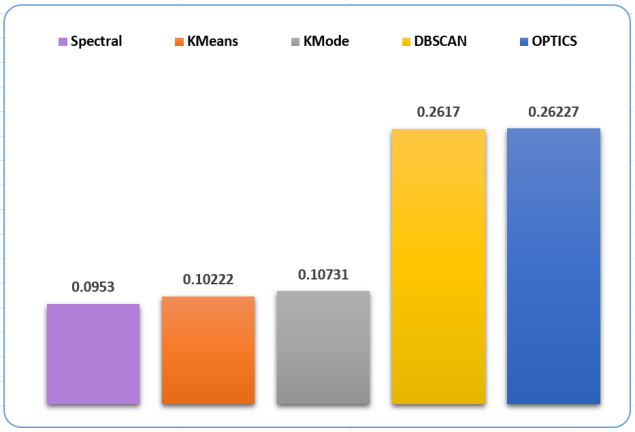


FIGURE 21. Validation results for Adjusted Mutual Information.

the algorithms across the validation charts (Fig. 16 – Fig. 18). We can observe K-means and K-modes generally achieved a slightly better outcome than OPTICS, while DBSCAN and Spectral a further behind. Clusters produced by Spectral were judged as below par by all the Internal validation metrics.

Now, to have a complete picture, a range of External validation techniques will be explored.

2) EXTERNAL VALIDATION

In this section, we will examine how well the clusters have been formed using external validation metrics.

a: ADJUSTED RAND INDEX (ARI)

The Rand Index (RI) determines a similarity score between two sets of clustering by considering each of the pairs of the provided samples and summing up pairs that are assigned in identical or different clusters in the *true clustering* in addition to the predicted ones [53]. The raw RI score is then 'adjusted for chance' into the ARI using (15). Scores towards +1 indicate better clustering.

$$ARI = \frac{\boldsymbol{RI} - \boldsymbol{Expected\_RI}}{\boldsymbol{max(RI)} - \boldsymbol{Expected\_RI}} \tag{15}$$

The index as depicted in Fig. 20, clearly projects the superior quality of clusters produced by OPTICS and DBSCAN, whereas the rest, according to ARI, did not show any commendable outcome.

b: ADJUSTED MUTUAL INFORMATION (AMI)

The AMI metric quantifies the extent of information the two clusters under examination have in common; often referred as 'Correlation Measure'. The MI score is then 'adjusted for chance' to get the AMI [54]. The AMI of two clusters $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$, is found using (16), where $\mathbf{T}$ is the Entropy. Scores approaching +1 indicate better clustering.

$$AMI(C_1, C_2) = \frac{[\boldsymbol{MI}\ (C_1, C_2) - \boldsymbol{Expected}\ (\boldsymbol{MI}\ (C_1, C_2))\ ]}{[\boldsymbol{avg}\ (\boldsymbol{T}\ (C_1)\ , \boldsymbol{T}\ (C_2)) - \boldsymbol{Expected}\ (\boldsymbol{MI}\ (C_1, C_2))]} \tag{16}$$

Figure 21 picturises the outcome for AMI. Algorithms other than DBSCAN and OPTICS have done better than according to the ARI balidation, but are still not close to the performance of OPTICS and DBSCAN.

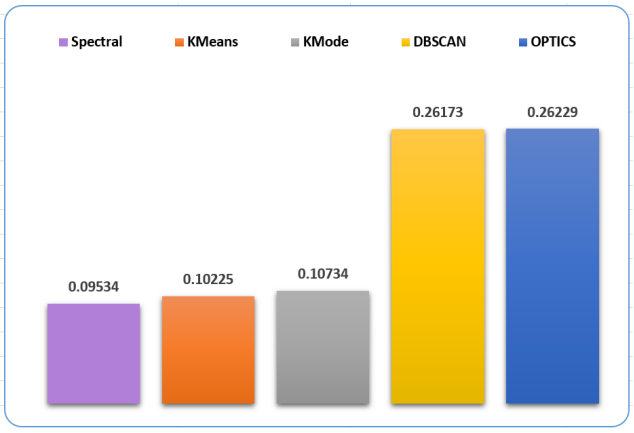


FIGURE 22. Validation results for V-measure.

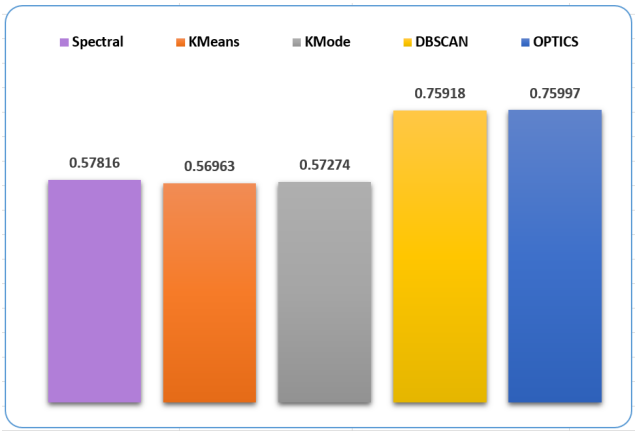


FIGURE 23. Validation results for Fowlkes-Mallows Index.

c: V-MEASURE

V-measure relies on the principle of conditional entropy analysis which is a measurement of the extent of disorder within the cluster. V-measure works out the Harmonic Mean of *Completeness* – whether all of the members of the class in question are allocated to the same cluster and *Homogeneity* – a measure of a cluster containing only members of a specific cluster [55]. Homogeneity and Completeness are two critical characteristics of a cluster. V-measure, $\boldsymbol{v}$ is given in (17). $\boldsymbol{\beta}$ signifies the degree of weightage given to each of these two characteristics, and in this case it is '1' (equal weightage).

$$\boldsymbol{v} = \frac{(\mathbf{1}+\boldsymbol{\beta})\times\text{completeness}\times\text{homogeneity}}{(\boldsymbol{\beta}\times\text{completeness}+\text{homogeneity})} \tag{17}$$

Results of the V-measure shown in Fig. 22 demonstrate the similarity to AMI with OPTICS and DBSCAN having better results than the rest.

d: FOWLKES-MALLOWS INDEX (FMI)

Another widely accepted external validation metric is the Fowlkes-Mallows Index (FMI). If $\boldsymbol{\alpha}$ is the percentage of correctly clustered data points, and $\boldsymbol{\beta}$ is the percentage of data points (in pairs) correctly allocated in the same cluster, then their *geometric mean*, $\boldsymbol{\mu}$, will account for FMI [56], i.e. $\boldsymbol{\mu} = \sqrt{\boldsymbol{\alpha}.\boldsymbol{\beta}}$.

FMI validates OPTICS and DBSCAN to be the top performers similar to the previous indices. However, the FMI also indicates that the clustering quality of other algorithms not much less than OPTICS and DBSCAN.

e: PURITY

Purity is a transparent and straightforward external validation metric. Often regarded as the 'Cluster Accuracy', t it has gained widespread acceptance as an indicator of the quality of the clusters produced by clustering algorithms. Scores approaching towards +1 suggest better clustering [57]. Fig. 21 charts the measures of purity for each algorithm.

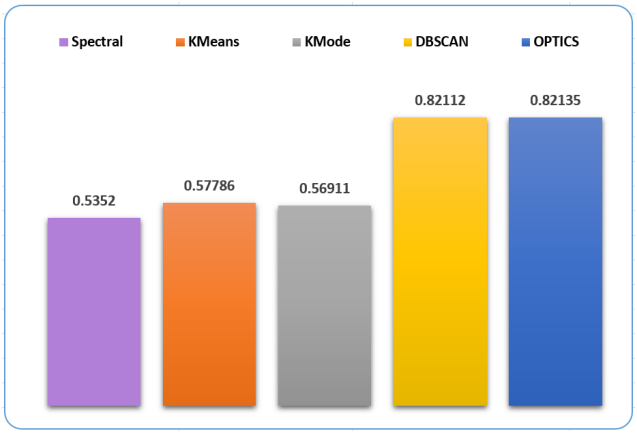


FIGURE 24. Validation results for purity.

ALGORITHM	∑ POSITION -1	∑ POSITION -2	∑ POSITION -3	∑ POSITION -4	∑ POSITION -5
Spectral	-	-	1	-	4
K-mode	-	-	1	4	-
K-means	-	-	3	1	1
DBSCAN	-	5	-	-	-
OPTICS	5	-	-	-	-

FIGURE 25. Summarised view of external validation outcomes.

f: SUMMARISING THE EXTERNAL VALIDATION OUTCOMES

Figure 25 has a summary of the findings of the external performance metrics that have been used in this research.

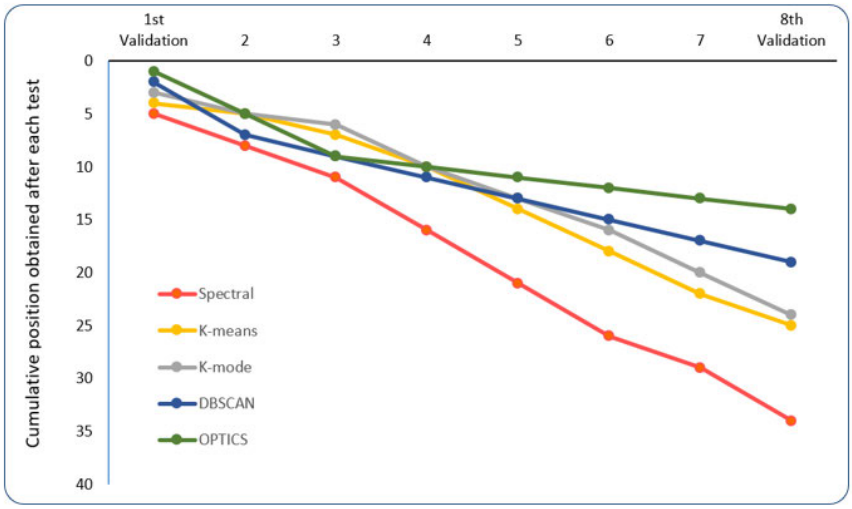


FIGURE 26. Cumulative positions after each validation tests.

	Spectral	*K-mode*	*K-means*	*DBSCAN*	*OPTICS*
Purity	0.5352	0.56911	0.57786	0.82112	0.82163
V-measure	0.09534	0.10734	0.10225	0.26173	0.26229
Adjusted Rand Index	-0.06834	-0.0028	0.00596	0.38781	0.38842
Fowlkes-Mallows Index	0.57816	0.57274	0.56963	0.75918	0.75997
Calinski-Harabasz Index	0.50764	0.51543	0.51551	0.50641	0.50643
Silhouette Coefficient Score	0.19172	0.29087	0.29054	0.12228	0.122259
Adjusted Mutual Information	0.0953	0.10731	0.10222	0.2617	0.26227
Reverse Davies-Bouldin Index	-0.34318	0.29521	0.26501	0.60158	0.60967

FIGURE 27. Heatmap of validation outcomes.

OPTICS has the top performance in all the external evaluation measures, while DBSCAN was a close second. The other three algorithms produced clusters of lesser quality. In fact, the trend depicted by the external metrics are the same across all the five evaluations techniques.

3) DISCUSSION ON THE INTERNAL AND EXTERNAL VALIDATION OUTCOMES

The figure below (Fig. 26) depicts the cumulative positions of each of the clustering algorithms, from Davies-Bouldin Index to the Purity metric.

Fig. 26 clearly shows that OPTICS had an uncontested clustering 'goodness' for the most part of the tests; while DBSCAN was relatively close.. The remaining three, failed to provide comparable performance to the top two, OPTICS and DBSCAN, though K-modes seemed occasionally promising. Based on all validation metrics' scores, on an average, clusters delivered by OPTICS had approximately a **0.26%** better performance than DBSCAN. It can be concluded that both OPTICS and DBSCAN produced meaningful clusters that align strongly with the aim of this research.

	Accuracy
OPTICS	0.75789
DBSCAN	0.75711
K-means	0.66739
K-mode	0.66658
Spectral	0.58417

FIGURE 28. Reported accuracy.

A Heatmap is generated in Fig. 27 to show the clusters created in terms of all the evaluation metrics in an easy-to-comprehend visualisation.

4) EVALUATION BASED ON CLASS DETECTION RATE

The validation techniques and scatterplots discussed so far, impart a convincing view on the clustering performance of the algorithms. In this section a more granular approach will be examined to investigate how well each of the clusters have

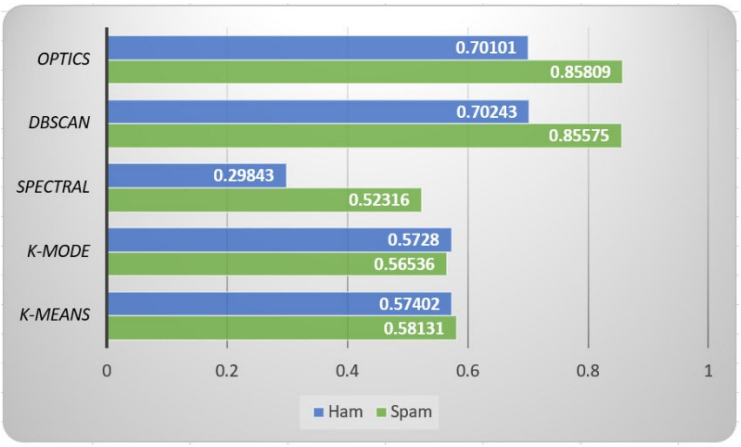


FIGURE 29. F-Scores achieved by the algorithms.

TABLE 7. A comparison with some relevant studies.

	approach	*results reported*	*additional comments*
Basavaraju *et al.* (2010) [10]	Experimental data were represented using Vector Space Model and a combination of BIRCH [59] and KNN was used for clustering	Accuracy => Around 74%	Major drawback is the size of the dataset which is of just 400 records. Does not consider analyzing character variations (often used by the spammers) as a feature for the proposed model.
Halder *et al.* (2011) [14]	Clustered using K-means and Expectation Maximization (EM). A combinational approach of Stylistic and Semantic features was investigated	Purity => 80% (K-means) and 57.4% (EM)	Dataset is limited with only 2600 records. Only email content is analyzed. No other extension of the work is proposed as a future improvement.
Cabrera-León *et al.* (2016) [15]	A proposition using Self-Organizing Map and Artificial Neural Network with relevant text preprocessing and topic based grouping.	Accuracy => Around 92.4%	Again the Dataset employed is limited in number and the framework shows compromised accuracy on email for topics that are not part of the experiment.
Mohammad (2011) [33]	A clustering proposition using Fuzzy C-Mean [60] algorithm. Involves Feature Extraction and Content Preprocessing.	False Positive Rate => 1.5%	Employs a large number of content based features, but does not clarify whether all of the features have been used and discussion on the dataset is quite restricted too.
This Study		The Average *Accuracy* (Balanced) for DBSCAN and OPTICS => 75.76% The Average *Purity* for DBSCAN and OPTICS => 82.12%	

been formed irrespective of the other through the application of metrics like *Accuracy* and *F-Score* [58]. Due to the availability of the original class labels, we can further evaluate the clusters using these two metrics.

Figure 28 shows the *Balanced Accuracy* achieved by each of the algorithms, where OPTICS and DBSCAN demonstrated acceptable performance. This is an overall evaluation that confirms the findings of the previous validation procedures.

Figure 29 reports the F-Scores, for both ham and spam emails separately. It is evident that OPTICS and DBSCAN performed better than the other algorithms. In addition the

TABLE 8. An extract of spam words and phrases from [27].

action	ad	bank	amazing	billion
celebrity	casino	beneficiary	bonus	billing
cialis	clearance	claim	deposit	debt
dream	earn	free	foreign	expire
xanax	your business	gift certificate	herbal	guarantee
unsolicited	low interest rate	work from home	online pharmacy	no investment
teen	free money	order now	instant access	million dollar

F-Scores for spam emails are higher (though not that impressive in most instances) than for ham in all cases except for K-modes, where the F-Score for ham is slightly higher than for spam emails. Generally F-Scores are good at indicating how precise and robust the individual clustering (*classifiers* for supervised model) are. The critical takeaway from the analysis of this section is that, though the proposition delivers encouraging results and interesting findings, there is still room for improvement in clustering quality, especially for Ham.

5) A COMPARISON WITH CLOSELY RELATED STUDIES

As tabulated in Table 7, our research has been compared to some of the closely related works that have used, at least partially, the concept of unsupervised learning in the segregation of ham and spam emails. Note that, as has been mentioned before, the availability of works that are largely similar to ours, is virtually nil. As described above, DBSCAN and OPTICS demonstrated strong performance > However, we found almost no relevant research that investigated any mainstream clustering algorithms except for K-means. It is therefore difficult to draw a decisive conclusion from the comparison.

VIII. CONCLUSION AND FUTURE WORK

This research described a novel framework based entirely on unsupervised methodologies to separate ham from spam emails through unsupervised clustering.

The process started with the formation of a raw dataset of several features of varying characteristics, based on the email's body content and subject header. Some of these features have not been used in earlier research with similar aims. Some features were completely novel and engineered after carefully examining the content from various angles; while other features used in this research have also been used in earlier research in a different form. The dataset was then converted into binary form and feature selection algorithms (MCFS and Laplacian) were introduced to remove low-impact features if possible through feature reduction algorithms. The feature that contributed the least was identified and consequently left out. The resulting dataset (containing both ham and spam emails) is now publicly available to download from github [62]. We believe this dataset can be a useful source for other relevant research.

Afterwards a range of unsupervised algorithms- K-means, K-modes, Spectral, DBSCAN and OPTICS were used to cluster the dataset to create clusters of ham and spam emails. An number of internal and external validation processes were then applied to the clusters to measure and quantify the quality and usefulness. The findings show that OPTICS and DBSCAN produce the best quality clusters, whereas the other algorithms, though some displayed sporadic promise, were not optimal. The clusters were further evaluated through metrics such as F-Score.

Differentiating ham from spam emails using 'only' unsupervised methods, acting upon the email's 'subject' field and its body content, is a novel approach. A range of novel features have also been engineered and sound feature selection methods have been applied to use only the features having sufficient degree of impact. Overall a number of novel avenues have been explored to address the significant research gaps in this field of study.

In future endeavors, we aim to further work on the novel features to improve the quality of clustering, especially for ham. We also intend to combine the framework presented here with our earlier work that deals with unsupervised clustering using Header fields only. This will provide us with a completely unsupervised system that can be expected to increase the detection rates and clustering quality using all the major parts of an email. The best performing clustering algorithms from this study and our earlier studies will be used in future work. The results will be validated using all standard evaluation processes available. Our datasets will also be made publicly downloadable for further use.

APPENDIX A

Currency coded words for feature f_7:

USD, US$, EURO, GBP, AUD, AU$, CAD, CA$, U$, JPY.

APPENDIX B

Stopwords removed from spaCy's default list:

{first, hundred, yourselves, yourself, our, out, nothing, amount, into, few, you, always, for, while, your, no, give, why, not, more, how, she, one, most, off, only, name, may, at, thus, regarding, please, again, call}

APPENDIX C

An ***extract*** of spam words and phrases from [27] has been shown in Table 8.

REFERENCES

[1] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261–168295, 2019, doi: 10.1109/access.2019.2954791.

[2] E. Bauer. *15 Outrageous Email Spam Statistics That Still Ring True in 2018, RSS*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.propellercrm.com/blog/email-spam-statistics

[3] D. Lynkova. (2021). *How Many Emails are Sent Per Day: The Startling Truth [2021], TechJury*. Accessed: Sep. 23, 2020. [Online]. Available: https://techjury.net/blog/how-many-emails-are-sent-per-day/

[4] K. Sheridan. (2020). *FBI: Business Email Compromise Cost Businesses 1.7B in 2019, Dark Reading*. Accessed: Mar. 21, 2021. [Online]. Available: https://www.darkreading.com/fbi-business-email-compromise-cost-businesses-17b-in-2019/d/d-id/1337035

[5] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5019–5081, Oct. 2020, doi: 10.1007/s10462-020-09814-9.

[6] J. Johnson. (2021). Spam e-mail: Countries of origin 2020. Statista. accessed: Nov. 27, 2020. [Online]. Available: https://www.statista.com/statistics/263086/countries-of-origin-of-spam/

[7] A. Karim, S. Azam, B. Shanmugam, and K. Kannoorpatti, "Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework," *IEEE Access*, vol. 8, pp. 154759–154788, 2020, doi: 10.1109/access.2020.3017082.

[8] O. Alonso, "Challenges with label quality for supervised learning," *J. Data Inf. Qual.*, vol. 6, no. 1, pp. 1–3, Mar. 2015.

[9] S. Manlangit, "Novel machine learning approach for analyzing anonymous credit card fraud patterns," *Int. J. Electron. Commerce Stud.*, vol. 10, no. 2, pp. 175–202, Dec. 2019, doi: 10.7903/ijecs.1732.

[10] M. Basavaraju and D. R. Prabhakar, "A novel method of spam mail detection using text based clustering approach," *Int. J. Comput. Appl.*, vol. 5, no. 4, pp. 15–25, Aug. 2010, doi: 10.5120/906-1283.

[11] R. M. Ravindran and D. A. S. Thanamani, "K-means document clustering using vector space model," *Bonfring Int. J. Data Mining*, vol. 5, no. 2, pp. 10–14, Jul. 2015, doi: 10.9756/bijdm.8076.

[12] C. Laorden, X. Ugarte-Pedrero, I. Santos, B. Sanz, J. Nieves, and P. G. Bringas, "Study on the effectiveness of anomaly detection for spam filtering," *Inf. Sci.*, vol. 277, pp. 421–444, Sep. 2014, doi: 10.1016/j.ins.2014.02.114.

[13] R. D. Kortum, "Hyperonyms and hyponyms," in *Varieties of Tone*. London, U.K.: Palgrave Macmillan, 2013, pp. 178–180, doi: 10.1057/9781137263544_23.

[14] S. Halder, R. Tiwari, and A. Sprague, "Information extraction from spam emails using stylistic and semantic features to identify spammers," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2011, pp. 104–107, doi: 10.1109/iri.2011.6009529.

[15] Y. Cabrera-León, P. G. Báez, and C. P. Suárez-Araujo, "Self-organizing maps in the design of anti-spam filters—A proposal based on thematic categories," in *Proc. 8th Int. Joint Conf. Comput. Intell.*, 2016, pp. 21–32, doi: 10.5220/0006041400210032.

[16] H. Padhiyar and P. Rekh, "An improved expectation maximization based semi-supervised email classification using Naïve Bayes and K-nearest neighbor," *Int. J. Comput. Appl.*, vol. 101, no. 6, pp. 7–11, Sep. 2014, doi: 10.5120/17689-8652.

[17] D. Hao, L. Zhang, J. Sumkin, A. Mohamed, and S. Wu, "Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2701–2710, Sep. 2020, doi: 10.1109/jbhi.2020.2974425.

[18] F. Qian, A. Pathak, Y. C. Hu, Z. M. Mao, and Y. Xie, "A case for unsupervised-learning-based spam filtering," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 367–368, Jun. 2010, doi: 10.1145/1811099.1811090.

[19] G. Dougherty, "Unsupervised learning," in *Pattern Recognition and Classification*. New York, NY, USA: Springer, 2012, pp. 143–155, doi: 10.1007/978-1-4614-5323-9_8.

[20] S. Russell and P. Norvig, "A modern, agent-oriented approach to introductory artificial intelligence," *ACM SIGART Bull.*, vol. 6, no. 2, pp. 24–26, Apr. 1995, doi: 10.1145/201977.201989.

[21] V. Starovoitov, "A clustering technique based on the distance transform," *Pattern Recognit. Lett.*, vol. 17, pp. 231–239, Mar. 1996, doi: 10.1016/0167-8655(95)00120-4.

[22] C.-C. Chang, J.-S. Chou, and T.-S. Chen, "An efficient computation of Euclidean distances using approximated look-up table," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 4, pp. 594–599, Jun. 2000, doi: 10.1109/76.845004.

[23] B. Guenter. *Spam Collection*. Accessed: Mar. 23, 2021. [Online]. Available: http://untroubled.rg/spam/

[24] *TREC Spam Collection*. Accessed: Mar. 23, 2021. [Online]. Available: https://trec.nist.gov/data/spam.html

[25] *ENRON Email Corpus*. Accessed: Mar. 29, 2021. [Online]. Available: https://www.cs.cmu.edu/~enron/

[26] *Hillary Clinton's Email*. Accessed: Apr. 15, 2021. [Online]. Available: https://www.kaggle.com/kaggle/hillary-clinton-emails/discussion/16419

[27] T. Freeman. *SpamTokens*. Accessed: Apr. 16, 2021. [Online]. Available: https://zenodo.org/record/4415744

[28] R. Tatman. *Fraudulent E-Mail Corpus*. Accessed: Apr. 18, 2021. [Online]. Available: https://www.kaggle.com/rtatman/fraudulent-email-corpus

[29] *spaCy*. Accessed: Jun. 2, 2021. [Online]. Available: https://spacy.io/

[30] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic text on social media," *Heliyon*, vol. 7, no. 2, Feb. 2021, Art. no. e06191, doi: 10.1016/j.heliyon.2021.e06191.

[31] *Accessing Text Corpora and Lexical Resources, Natural Language Toolkit—NLTK 3.5 Documentation*. Accessed: Jan. 21, 2021. [Online]. Available: http://www.nltk.org/book_1ed/ch02.html

[32] W. Wu. *Efficient MinHash-Based Algorithms for Big Structured Data*. Accessed: Dec. 17, 2020. [Online]. Available: https://opus.lib.uts.edu.au/bitstream/10453/128013/1/01front.pdf

[33] N. T. Mohammad, "Fuzzy clustering approach to filter spam e-mail," in *Proc. World Congr. Eng.*, vol. 3, 2011, pp. 1–6.

[34] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 27, no. 1, pp. 46–57, Jan. 2015, doi: 10.1016/j.jksuci.2014.03.014.

[35] R. Liu, N. Yang, X. Ding, and L. Ma, "An unsupervised feature selection algorithm: Laplacian score combined with distance-based entropy measure," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, 2009, pp. 65–68, doi: 10.1109/iita.2009.390.

[36] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2010, pp. 333–342, doi: 10.1145/1835804.1835848.

[37] X. Zhu, Y. Wang, Y. Li, Y. Tan, G. Wang, and Q. Song, "A new unsupervised feature selection algorithm using similarity-based feature clustering," *Comput. Intell.*, vol. 35, no. 1, pp. 2–22, Feb. 2019, doi: 10.1111/coin.12192.

[38] J. Sander, "Density-based clustering," in *Encyclopedia of Machine Learning and Data Mining*. New York, NY, USA: Springer, 2016, pp. 1–5.

[39] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conf. Series, Mater. Sci. Eng.*, vol. 336, Apr. 2018, Art. no. 012017, doi: 10.1088/1757-899x/336/1/012017.

[40] V. R. Patel and R. G. Mehta, "Data clustering: Integrating different distance measures with modified k-means algorithm," in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011* (Advances in Intelligent and Soft Computing). New Delhi, India: Springer, 2012, pp. 691–700, doi: 10.1007/978-81-322-0491-6_63.

[41] J. Liu and J. Han, "Spectral clustering," in *Data Clustering*. USA: Chapman & Hall, 2018, pp. 177–200, doi: 10.1201/9781315373515-8.

[42] H. Zhou, Y. Zhang, and Y. Liu, "A global-relationship dissimilarity measure for the k-modes clustering algorithm," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–7, Mar.'2017, doi: 10.1155/2017/3691316.

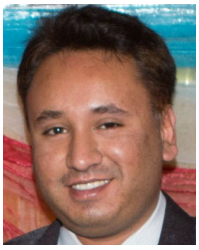
[43] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1999, pp. 49–60, doi: 10.1145/304182.304187.

[44] G. S. Semaan, A. C. Fadel, J. A. D. M. Brito, and L. S. Ochi, "A hybrid heuristic with Hopkins statistic for the automatic clustering problem," *IEEE Latin Amer. Trans.*, vol. 17, no. 1, pp. 7–17, Jan. 2019, doi: 10.1109/tla.2019.8826689.

[45] C. Albon, *Machine Learning With Python Cookbook: Practical Solutions From Preprocessing to Deep Learning*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[46] L. J. Deborah, R. Baskaran, and A. Kannan, "A survey on internal validity measure for cluster validation," *Int. J. Comput. Sci. Eng. Surv.*, vol. 1, no. 2, pp. 85–102, Nov. 2010, doi: 10.5121/ijcses.2010.1207.

[47] S. Gajawada and D. Toshniwal, "Hybrid cluster validation techniques," in *Advances in Computer Science, Engineering & Applications* (Advances in Intelligent Systems and Computing). Berlin, Germany: Springer, 2012, pp. 267–273, doi: 10.1007/978-3-642-30111-7_25.
[48] M. Mughnyanti, S. Efendi, and M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with Davies–Bouldin index evaluation," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 725, Jan. 2020, Art. no. 012128, doi: 10.1088/1757-899x/725/1/012128.
[49] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi: 10.1109/tpami.1979.4766909.
[50] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 911–916, doi: 10.1109/icdm.2010.35.
[51] X. Wang and Y. Xu, "An improved index for clustering validation based on Silhouette index and Calinski–Harabasz index," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 569, no. 5, 2019, Art. no. 052024, doi: 10.1088/1757-899x/569/5/052024.
[52] H. B. Zhou and J. T. Gao, "Automatic method for determining cluster number based on Silhouette coefficient," *Adv. Mater. Res.*, vol. 951, pp. 227–230, May 2014, doi: 10.4028/www.scientific.net/amr.951.227.
[53] R. R. de Vargas and B. R. C. Bedregal, "A way to obtain the quality of a partition by adjusted Rand index," in *Proc. 2nd Workshop-School Theor. Comput. Sci.*, Oct. 2013, pp. 67–71, doi: 10.1109/weit.2013.33.
[54] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1073–1080, doi: 10.1145/1553374.1553511.
[55] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: Review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, Jun. 2019, Art. no. e01802, doi: 10.1016/j.heliyon.2019.e01802.
[56] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings: Rejoinder," *J. Amer. Stat. Assoc.*, vol. 78, no. 383, p. 584, Sep. 1983, doi: 10.2307/2288123.
[57] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2012.
[58] D. Hand and P. Christen, "A note on using the F-measure for evaluating record linkage algorithms," *Statist. Comput.*, vol. 28, no. 3, pp. 539–547, May 2018, doi: 10.1007/s11222-017-9746-6.
[59] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 1996, pp. 103–114, doi: 10.1145/233269.233324.
[60] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, 1984, doi: 10.1016/0098-3004(84)90020-7.
[61] *LigSpam*. Accessed: May 19, 2021. [Online]. Available: https://www.stat.purdue.edu/~mdw/598/datasets.html
[62] *Dataset Used in This Study*. Accessed: Jul. 11, 2021. [Online]. Available: https://github.com/Asif5566/dataextract/blob/master/DataSet.zip



ASIF KARIM (Member, IEEE) is currently a Research Active Lecturer with Charles Darwin University, Australia. He has considerable industry experience in IT, primarily in the field of software engineering. His research interests include machine intelligence, health informatics, cyber security, and smart contracts.

SAMI AZAM (Member, IEEE) is currently a Leading Researcher and a Senior Lecturer with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He is actively involved in the research fields relating to computer vision, signal processing, artificial intelligence, and biomedical engineering. He has number of publications in peer reviewed journals and international conference proceedings.

BHARANIDHARAN SHANMUGAM is currently a Research Intensive Senior Lecturer with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He has a large number of publications in several different journals and conference proceedings. His research interest includes cybersecurity.

KRISHNAN KANNOORPATTI is currently a Research Active Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. In addition of being a Stellar Academic and an Innovative Researcher, he also has an extensive experience of working with the government bodies in setting up data privacy policies at national and state level.