Leveraging LDA and TF-MONO to Enhance Spam Email Categorization

Janice Jane C. Floreta
College of Computing and Information Sciences, Caraga State University

Frances Ruth M. Magayon
College of Computing and Information Sciences, Caraga State University

Leo C. Magallanes
College of Computing and Information Sciences, Caraga State University

Cristopher C. Abalorio
College of Computing and Information Sciences, Caraga State University

他

https://doi.org/10.5109/7395503

出版情報：Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 11, pp.93-99, 2025-10-30. International Exchange and Innovation Conference on Engineering & Sciences
バージョン：

Leveraging LDA and TF-MONO to Enhance Spam Email Categorization

Janice Jane C. Floreta[1], Frances Ruth M. Magayon[2], Leo C. Magallanes[3], Cristopher C. Abalorio[4], Regien B. Bakazato[5], Roland P. Abao[6], Vicente A. Pitogo[7]
[1,2,3,4,5,6,7]College of Computing and Information Sciences, Caraga State University – Main Campus, Butuan City, Philippines
ccabalorio@carsu.edu.ph

Abstract: *In this study, TF-MONO classification performance is compared with TF-IDF using three classifiers: Naive Bayes, Decision Tree, and Random Forest. Determined the optimal alpha value for TF-MONO, finding it to be 8.0 and 9.0, with Naive Bayes performing best at these values. Results show that Random Forest performed better with low alpha values (5.0 and 6.0) while Decision Tree is non-parametric as its accuracy does not change regardless of various alpha (balance parameters) used. Decision Tree and Random Forest also perform better with TF-MONO compared to TF-IDF. In addition, Naive Bayes achieved the highest accuracy (84%) with TF-MONO, outperforming TF-IDF (65%). It showed that TF-Mono performs better than TF-IDF. This study highlights converted binary-class dataset into multiclass datasets increasing the accuracy and efficiency of email spam categorization by addressing the evolving challenges of spam detection in the digital era.*

Keywords: TF-MONO; LDA; Multiclass Classification; Spam Detection.

1. INTRODUCTION

Email has been a fundamental communication medium since the Internet's early days, with many emails sent and received daily [1]. An increasing number of internet users has made email spam a significant concern nowadays [2]. Spam is a prevalent issue that may impact someone who uses email. It is an unsolicited, mass email sent for a variety of purposes, such as phishing, fraud, advertising, and more, by unknown senders. Commercial ads or viruses that might damage users' computers and data are frequently found in spam communications [3].

As a result of an increase in email usage, spam, worms, and viruses have also grown over time. Email fraud and spam are serious problems for Internet consumers as well as businesses of all kinds. According to Kaspersky Lab, spam comprised 56.51% of total email traffic in 2019, increasing 4.03 percentage points from 2018, with China responsible for 21.26%. Russia's spam market share increased from 21.27% in 2020 to 50.37% in 2022, while spam made up 48.63% of global emails, with Russia being a major source at 29.82%. In addition, Statistica noted that in 2023, the U.S. set a record by sending the most spam emails in a day, approximately eight billion [4][5].

To tackle the issue of spam, various methods like text classification in the category of spam detection particularly in categorization. One crucial stage in text classification is term weighting. Term weighting schemes assign appropriate weight to the terms. Term weighting schemes (TWS) are classified into unsupervised and supervised schemes. Over time, term weighting with a supervised approach has consistently outperformed unsupervised methods in terms of classification performance. This is because supervised methods can learn from labeled data, which provides them with a better understanding of the relationships between terms and categories [6][7][8]. Supervised Term Weighting Scheme is preferred because it beats the text classification performance of the preceding term weighting scheme strategies [9].

In the earlier days, spam detection approaches mainly focused on the approaches where they calculated the importance of given words in certain documents and so used weighting schemes like Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), and TF-IGM (term frequency and inverse gravity moment), which are very popular in text classification [10]. However, these methods, while foundational, often fail to capture spam messages' nuanced and evolving nature, which can vary significantly in content and presentation. Existing spam detection uses benchmark datasets with only two categories, spam and ham, usually binary classification. Ham refers to legitimate or non-spam messages in the context of email [11]. However, actual spam emails manifest in many forms or categories. These occurring contents could also be included as categories to perform classification better. Henceforth, binary classification can now be extended to multiclass classification.

This study suggests a supervised Term Weighting Scheme the TF-MONO (Term Frequency Max Occurrence and Non-Occurence) method. TF-MONO is generally used in text classification which has many subdomains according to their contents, and predefined categories like Author recognition [12] sentiment analysis [13][14][15], and short text classification [16] can be displayed in the available literature of text classification as several sub-domains and has shown its effectiveness. However, in Supervised Term Weights have not yet been explored particularly for Spam Email Document Categorization and in Spam Email Detection. Through the use of TF-MONO in which it is for numeral representation and utilizing Latent Dirichlet Allocation (LDA) for topic modeling in identifying the topic of the collection of documents within the dataset of email spam. Moreover, it converts a binary-class dataset into a multiclass dataset, increasing the accuracy of email spam categorization. Furthermore, in TF-MONO formula

Alpha (α) is a variable that determines the range of global weight values throughout the weighting period its defined value ranges 5.0–9.0 and default value 7.0, respectively [17].

Overall, this study contributes to the ongoing efforts to combat spam and aims to leverage methods to enhance the accuracy of the classifier, to classify a label. In addition, it aims to give essential insights into the furtherance of more adaptable spam detection algorithms and explores the potential of enhancing spam email document categorization.

2. REVIEW OF RELATED LITERATURE

2.1 Spam Detection

Spam detection is the process of recognizing and filtering unknown messages, which usually appear in electronic communications like emails, text messages, or website comments. Spam may take different forms, including ads, phishing attempts, virus dissemination, and irrelevant or undesirable information. To combat spam, detection systems employ multiple techniques to differentiate between legitimate and spam messages [18].

2.2 LDA Technique

The LDA (Latent Dirichlet Allocation) model of a corpus is regarded as a generative model in the probabilistic sense LDA, [19] to begin with, assumes a process of document making through several topical stages. Then the topics generate the words contained in the documents with certain probabilities. For a given set of documents, LDA inverts this process and tries to find the latent topics that would give rise to such documents.

2.3 Multi-class Classification

Multiclass classification systems are crucial for categorizing emails into specific spam subcategories, enhancing security measures and user protection. Techniques like Latent Dirichlet Allocation (LDA) and Term Frequency - Max-Occurrence and Non-Occurrence (TF-MONO) are explored for their potential in improving spam email classification systems . While binary-class classification has been effective in filtering generic spam, multiclass systems offer a more detailed approach, enabling better identification of various types of spam such as fraudulent, threatening, and suspicious emails [20]. Additionally, machine learning algorithms like Naive Bayes and neural networks have shown promise in accurately classifying spam emails, with neural networks achieving high accuracy rates even with limited input features [21]. Continuous advancements in classification models are essential to combat evolving spam techniques and protect users effectively.

2.4 TF-MONO and TF-IDF

TF-MONO is a controlled and supervised scheme for term weighting that takes into account the frequency of terms, the most frequently used term, and also the absences of certain terms. Furthermore, it employs the information on how common terms are used within a certain group and their use in differing groups to improve the performance of the system that is designed to identify spam emails [22]. On the other hand, TF-IDF, a popular measure in information retrieval, evaluates term importance based on their frequency in documents relative to a document collection [23]. TF-MONO aims to enhance text classification accuracy by incorporating factors like maximum occurrence and non-occurrence in addition to term frequency [24]. In contrast, TF-IDF focuses on capturing the significance of terms by considering their frequency within the entire document collection [25]. While TF-MONO introduces a novel approach to supervised term weighting, TF-IDF is used to measure for assessing term importance in text classification tasks [26].

2.5 Related Studies

Spam detection has become increasingly complex as spammers develop new techniques to bypass traditional filters. This survey paper explores various methods to combat spam emails, including the Term Weighting Scheme (TWS) algorithm, which achieves high accuracy rates [18]. Machine learning approaches are also proving effective, with researchers constantly developing improved filtering systems [27].

Most of the existing methods of spam use machine learning techniques that are considered the most accurate and can classify spam as the highest accuracy. These are the most widely used classification algorithms selected to select the most efficient ones in detecting spam [28]. The following six commonly used machine learning classification algorithms are the following: Naive Bayes, Random Forest, Decision Tree, and Logistic Regression. Naive Bayes is a probabilistic algorithm that is effective at categorizing spam. One of the algorithm's drawbacks when dealing with spam emails is that the received email might contain a word that wasn't present in the training set, which would lower the classification quality [29]. SVM is a linear classifier that can be thought of as the maximum indentation equivalent of locating the hyperplane dividing the classes.

Logistic regression is a suitable analytical technique for creating a model of the data and explaining its relationship with binary response outcome and predictor variables. The outcome, which is restricted to values between 0 and 1, is the likelihood of allocating a value to a particular class [30]. A decision tree is a ranked structure that separates the feature space into subspaces and provides prediction for every object in each subspace and the algorithm often overfits. An algorithm for making predictions called Random Forest builds on the concept of trees which improves each tree's capacity for prediction on its own which is accomplished by considering the highest voted classes of each tree as the output. Overall, NB and logistic regression yield the highest accuracy, up to 99%. By combining the schemes or filtering approaches, the results can be used to develop a classifier for a detecting spam that is more intelligent [28].

3. METHODS

This section elaborates on how the study was done, from obtaining the Spam Email dataset to building the model, which can be shown in Fig. 1.

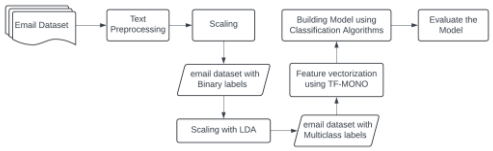

Fig. 1. Framework of the Study

3.1 Data collection

The email dataset used in this study, shown in Table 1, was the spam and ham mail dataset downloaded from Kaggle. It contained a collection of emails annotated with spam or ham (which means not a spam email). The email corpus consisted of 5170 datasets, with 71 percent classified as ham and 29 percent as spam. The collection of documents consisted of two distinct groups, namely the training set and the test set. The processing and building of the classifiers were done using the training set, while the classification performance assessment was conducted using the test set.

Table 1. Spam Email Dataset in Binary Labels

Label	Category	No. of document
0	ham	3671
1	spam	1499

3.2 Data Preprocessing

The dataset was prepared by involving data cleaning, removing stop-words, tokenization, and lemmatization. The preparation of data was essential for ensuring the data was in a suitable format and condition.

3.3 Scaling and Topic Modelling

The email dataset was a binary classification, and to transform it into a multiclass dataset, Latent Dirichlet Allocation was utilized. Scaling involved the reannotation of the email dataset to classify spam and ham using weights. The binary dataset was transformed into multiclass using topic modeling.

3.4 Multiclass Dataset

The researchers conducted multiclass spam detection research using TF-Mono, utilizing suitable multiclass datasets with labeled examples of spam and non-spam (ham) classes that were required. Multiclass classification in this task involved classifying into one of more than two classes or categories. After the labeling of spam and ham with 10 classes or categories—ham_label1, ham_label2, ham_label3, ham_label4, ham_label5, spam_label1, spam_label2, spam_label3, spam_label4, and spam_label5—as shown in Table 3-2 below.

3.5 Labeling Classes and LDA Topic Modeling

The email spam dataset was a binary classification, and to transform it into multiclass, it needed processing. First, the researchers used Latent Dirichlet Allocation from sklearn.decomposition for topic modeling to identify the topics in the dataset. After the researchers identified a topic, they created a new label for the categories "spam" and "ham" to a "New_Label" based on the topic. The researchers identified 10 new class labels in topic modeling. LDA topic modeling randomly assigned each word in each document to 10 topics (labels). The results were for randomly assigning each word to the documents. After randomly assigning each word to a topic, the researchers then created new labels for multiclass: ham_label1, ham_label2, ham_label3, ham_label4, and ham_label5 for the ham category, and spam_label1, spam_label2, spam_label3, spam_label4, and spam_label5 for the spam category. The results of the new labels based on the assigned topics were shown in Table 2. The column named "Topic Words" represented the results of topic modeling that randomly assigned each word to the documents of 10 topics (labels).

Table 2. Label Based on the Topics

New_Label	Topics Word
ham_label1	['elli', 'echidna', 'fuily', 'cayenne', 'association', 'command', 'helm', 'hub', 'brewery', 'dit', 'dramaturgy', 'bupropion', 'dia', 'awareness', 'ebony']
…….	…….
spam_label5	['corel', 'server', 'download', 'photoshop', 'mx', 'pro', 'office', 'cd', 'professional', 'microsoft', 'software', 'xp', 'adobe', 'window']

3.6 Text Pre-Processing Result

After the dataset was converted into multiclass, the next step was text preprocessing. Text preprocessing plays an essential role in in classification of text as it normalizes the text representation and produces more accurate and reliable classification results. Tokenization involves splitting the text into smaller units, called tokens. Stop words removal entails eliminating common English words that do not contribute meaning to a sentence or text in a document. Stemming reduces words to their root form, capturing the core meaning.

3.7 Feature Extraction and Vocabulary

After pre-processing the text in the document, the next step was feature extraction. A CountVectorizer was used for the feature extraction technique that is common in natural language processing to convert the text into a matrix count of tokens. Using the CountVectorizer, the text data was transformed into numerical values. The vocabulary in CountVectorizer represented a collection in a text document of all unique tokens or words that were extracted. The vocabulary mapped each unique token to an integer, counting all repeated words inside the document (e.g., 'gmail': 0, 'spam': 1, 'message': 2).

3.8 Feature Names

After the vocabulary, the next step is feature names to represent the unique word of tokens that are found in the text data. The feature names are derived from vocabulary during the process of CountVectorizer. In Fig. 2 is the visualization of showing the top 100 words feature name extracted using the CountVectorizer.

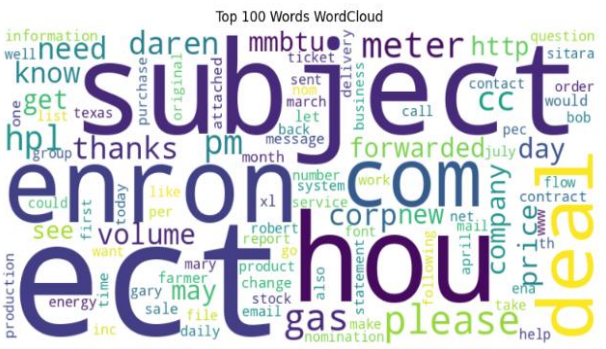

Fig. 2. Word Cloud Top 100 words feature names.

3.9 TF-MONO

After the feature extraction, the next step is the TF-Mono. The researchers used a new supervised term weighting scheme for text classification TF-Mono where the TF is Term Frequency and MONO, MO for Max-Occurence and NO for Non-Occurrence. The Max-Occurence (MO) is calculated by only getting the maximum of all document frequencies over the classes, 10 classes in a document have ham_label1, ham_label2, ham_label3, ham_label4, ham_label5, spam_label1, spam_label2, spam_label3, spam_label4 and spam_label5.

Table 3. MONO Global

Features	MONO Local	MONO Global
aa	0.0015	1.0123
aaa	0.0034	1.0272
aabda	0.0048	1.0383
aabvmmq	0.0030	1.0237
aac	0.0013	1.0102
aachecar	0.0051	1.0407
aaer	0.0015	1.0123
aafco	0.0013	1.0102
aaiabe	0.0000	1.0000
aaigrcrb	0.0000	1.0000
aaihmqv	0.0011	1.0091

In Table 3 is the result of MONOGlobal = [1 + α * MONOLocal]. As alpha (α) is the variable used to set the ranges of global weights values acting as a balance parameter. Its value ranges and default values are set as 5.0–9.0 and 7.0. MONOGlobal = [1 + α * MONOLocal].

4. RESULTS AND DISCUSSION

4.1 Performance Evaluation

After training, the model was evaluated for performance using the testing data. The researchers used a classification report for metrics, obtaining the precision, recall, F1 score, and accuracy of the Naive Bayes, Decision Tree, and Random Forest classifiers using TF-IDF and TF-MONO. A comparison of the two term weighting schemes was conducted to determine which of the three classifiers performed best, as well as to assess the balance parameter (α).

Table 4. Comparison of Accuracy Value of Balance Parameter (α)

	Balance Parameter (α)				
Classifier	**5.0**	**6.0**	**7.0**	**8.0**	**9.0**
Naïve Bayes	0.84	0.84	0.84	0.84	0.84
Decision Tree	0.72	0.72	0.72	0.72	0.72
Random Forest	0.80	0.80	0.79	0.79	0.79

The results in Table 4 presented the Alpha values of 5.0 and 6.0, with an average of 78.47%, and an Alpha value of 7.0, with an average value of 78.44%. The highest accuracy for the balance parameter (α) was observed at values of 8.0 and 9.0, with an average of 78.54%. Therefore, the researchers utilized the balance parameters of 8.0 and 9.0.

Table 5.Classification Report of Naive Bayes TF-MONO

Label	Precision	Recall	F1-score	Support
ham_label1	0.98	0.80	0.88	75
ham_label2	0.80	0.95	0.87	176
ham_label3	0.91	0.98	0.95	239
ham_label4	0.61	0.65	0.63	79
ham_label5	0.99	0.94	0.96	148
spam_label1	0.73	0.73	0.73	44
spam_label2	0.76	0.83	0.79	105
spam_label3	0.81	0.651	0.72	96
spam_label4	0.59	0.31	0.41	32
spam_label5	0.81	0.62	0.70	42
accuracy			0.84	1036
macro avg	0.80	0.75	0.76	1036
weighted avg	**0.84**	**0.84**	**0.83**	1036

The results of the classification report include the precision, recall, f1 score, and accuracy, in Naive Bayes Classifier of TF-IDF (shown in Table 6) and for TF-MONO (shown in Table 5). The accuracy of Naive Bayes in TF-MONO has 0.84 and TF-IDF has 0.69 which is TF-MONO is the highest accuracy for Naive Bayes classifier.

Table 6. Classification Performance of Naïve Bayes TF-IDF

Label	Precision	Recall	F1-score	Support
ham_label1	**1.00**	0.51	0.67	75
ham_label2	0.38	0.93	0.54	176
ham_label3	0.76	**0.97**	0.85	239
ham_label4	**1.00**	0.14	0.24	79
ham_label5	0.98	0.88	**0.93**	148
spam_label1	1.00	0.20	0.34	44
spam_label2	0.78	0.64	0.70	105
spam_label3	0.38	0.84	0.25	96
spam_label4	**1.00**	0.06	0.12	32
spam_label5	**1.00**	0.07	0.13	42
accuracy			0.65	1036
macro avg	**0.87**	**0.45**	**0.48**	1036
weighted avg	0.80	0.65	0.61	1036

Table 7. Classification Performance of Decision Tree TF-MONO

Label	Precision	Recall	F1-score	Support
ham_label1	0.84	**0.91**	**0.91**	156
ham_label2	0.68	0.54	0.55	79
ham_label3	0.85	0.65	0.68	78
ham_label4	0.56	0.79	0.64	48
ham_label5	**0.88**	0.70	0.69	94
spam_label1	0.50	0.55	0.59	92
spam_label2	0.64	0.47	0.61	40
spam_label3	0.64	0.65	0.62	186
spam_label4	0.38	0.85	0.83	231
spam_label5	0.58	0.29	0.35	31
accuracy			**0.72**	1036
macro avg	**0.65**	**0.65**	**0.64**	1036
weighted avg	**0.72**	**0.72**	**0.72**	1036

The results of the classification report include the precision, recall, f1 score, and accuracy, in Decision Tree Classifier of TF-IDF (shown in Table 8) and for TF-MONO (shown in Table 7). The accuracy of the Decision Tree in TF-MONO has 0.71 and TF-IDF has 0.70. TF-MONO is the higher accuracy for the Decision Tree classifier.

Table 8. Classification Performance of Decision Tree TF-IDF

Label	Precision	Recall	F1-score	Support
ham_label1	0.81	0.72	0.76	75
ham_label2	0.58	0.62	0.68	176
ham_label3	0.84	0.83	0.83	239
ham_label4	0.50	0.51	0.50	79
ham_label5	**0.93**	**0.95**	**0.94**	148
spam_label1	0.62	0.75	0.68	44
spam_label2	0.60	0.55	0.57	105
spam_label3	0.52	0.52	0.52	96
spam_label4	0.36	0.25	0.30	32
spam_label5	0.51	0.52	0.52	42
accuracy			0.69	1036
macro avg	0.63	0.62	0.62	1036
weighted avg	0.69	0.69	0.69	1036

Table 9. Classification Performance of Random Forest TF-MONO

Label	Precision	Recall	F1-score	Support
ham_label1	**0.98**	**0.96**	0.80	75
ham_label2	0.77	0.53	0.81	176
ham_label3	0.86	0.67	0.90	239
ham_label4	0.75	0.83	0.62	79
ham_label5	0.97	0.84	**0.96**	148
spam_label1	0.46	0.63	0.62	44
spam_label2	0.65	0.57	0.75	105
spam_label3	0.80	0.83	0.67	96
spam_label4	1.00	0.95	0.40	32
spam_label5	0.89	0.35	0.53	42
accuracy			0.79	1036
macro avg	0.81	0.70	0.71	1036
weighted avg	0.82	0.79	0.79	1036

The results of the classification report include the precision, recall, f1 score, and accuracy, in Random Forest Classifier of TF-IDF (shown in Table 10) and for TF-MONO (shown in Table 9). The accuracy of Random Forest in TF-MONO has 0.71 and TF-IDF has 0.70 which is TF-MONO is the higher accuracy for Random Forest classifier.

Table 10. Classification Performance of Random Forest TF-IDF

Label	Precision	Recall	F1-score	Support
ham_label1	0.98	0.68	0.80	75
ham_label2	0.62	0.86	0.72	176
ham_label3	0.89	0.92	0.91	239
ham_label4	0.73	0.41	0.52	79
ham_label5	**0.96**	**0.93**	**0.94**	148
spam_label1	0.74	0.84	0.79	44
spam_label2	0.63	0.86	0.73	105
spam_label3	0.78	0.66	0.71	96
spam_label4	0.82	0.28	0.42	32
spam_label5	0.94	0.40	0.57	42
accuracy			0.78	1036
macro avg	**0.81**	**0.68**	**0.71**	1036
weighted avg	**0.80**	**0.78**	**0.77**	1036

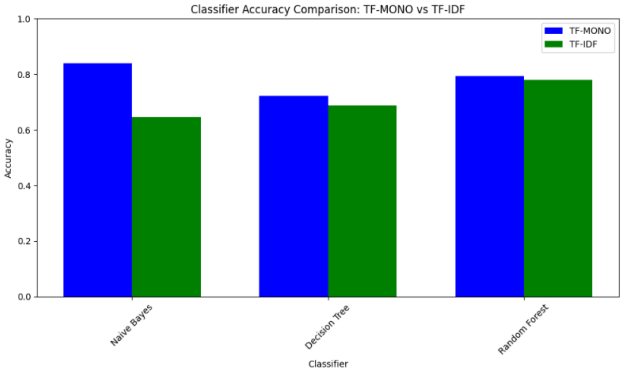

Fig. 3. Comparison of Accuracy: TF-MONO vs TF-IDF

5. SUMMARY AND CONCLUSION

5.1 Summary

This study investigates the supervised term weighting scheme TF-MONO and Multiclass scaling of binary classification through Latent Dirichlet Allocation for topic modeling for identifying the topic of the collection of documents within the dataset of email spam. Enhancing spam email document categorization using term weighting scheme TF-MONO. The researchers acquired a dataset of email spam and ham which was acquired on Kaggle.com. The researchers convert the dataset email spam binary into multiclass by using a Latent Dirichlet Allocation for topic modeling. Then the researchers pre-process the document using Tokenization, Stop Words, and Stemming. After the preprocessing the researchers extract the document by using Feature Extraction CountVectorizer.

The study's methodology of TF-MONO for enhancing spam email document categorization. Leveraging Latent Dirichlet Allocation for topic modeling to generate a topic from a document. The researchers first collect data from email spam. Data is then preprocessed using Tokenization, Stop Words, and Lemmatization then extracted to get feature names. A machine learning classifier, likely trained on a multiclass labeled data, then predicts the categorization.

5.2 Conclusion

To conclude, this study successfully explored the topic technique of Latent Dirichlet Allocation and TF-MONO. Multiclass scaling for email spam categorization. It aims to leverage these methods to enhance the accuracy of the classifier, to classify a label. The researchers employed a multiclass dataset from converting to binary, utilizing classification as multiclass in email spam categorization.

Among the three classifiers, Naive Bayes has the best result for using the alpha 8.0 and 9.0 obtained (84%) accuracy in TF-MONO while it obtained (65%) accuracy in TF-IDF. To achieve the result of high accuracy of the classifier of TF-MONO. The researchers makes a Comparison of TF-MONO and TF-IDF for which is the high accuracy of TF-IDF and TF-MONO. The 3 classifiers Naive Bayes, Decision Tree, and Random Forest. In computing the MONO Global first the researchers needed to compare which is the best alpha for the classifier. After computing all default values of alpha (5.0 to 9.0) the best alpha is 8.0 and 9.0 with the average of (78.54%) as shown in Table 4 - 11. Computing the alpha Naive Bayes is probabilistic is better with a high

alpha value 8.0 and 9.0, Decision Tree is non-parametric, that's why it does not matter of alpha value either low value or high value of alpha, and Random Forest is better with a low value of alpha 5.0 and 6.0. In the TF-MONO using the alpha 8.0 and 9.0 classifier Naive Bayes obtained (84%) accuracy while the TF-IDF obtained (65%) accuracy. The Classifier Decision Tree in TF-MONO obtained (72%) accuracy while TF-IDF obtained (69%) accuracy. Classifier Random Forest the TF-MONO obtained (79%) and TF-IDF obtained (78%). Among the 3 classifiers, Naive Bayes has an impressive accuracy of (84%). TF-MONO best fit with Naive Bayes classifier with a best alpha value of 8.0 and 9.0 in Computing of MONO Global. TF-Mono with the average accuracy of (78%) and TF-IDF with the average accuracy (70%). Indeed TF-Mono performs better than TF-IDF

6. REFERENCES

[1] N. M. Shajideen and B. V., "Spam Filtering: A Comparison Between Different Machine Learning Classifiers," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1919–1922. doi: 10.1109/ICECA.2018.8474778.

[2] 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, 2020, doi: 10.1109/ACCESS.2019.2954791.

[3] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine Learning Techniques for Spam Detection in Email and IoT Platforms : Analysis and Research Challenges," vol. 2022, 2022, doi: 10.1155/2022/1862888.

[4] T. Kulikova and T. Sidorina, "Spam and phishing in Q3 2020." https://securelist.com/spam-and-phishing-in-q3-2020/99325/

[5] T. Kulikova, O. Svistunova, A. Kovtun, I. Shimko, and R. Dedenok, "Spam and phishing in 2023." https://securelist.com/spam-phishing-report-2023/112015/

[6] Y. Zhang, M. Wang, F. Gottwalt, M. Saberi, and E. Chang, "Ranking scientific articles based on bibliometric networks with a weighting scheme," *J. Informetr.*, vol. 13, no. 2, pp. 616–634, 2019, doi: https://doi.org/10.1016/j.joi.2019.03.013.

[7] Y. Sakai, K. Takeuchi, and E. Ito, "A Study of Flaming Comment Detection using Text based machine learning," *Inf. Process. Soc. Japan*, vol. 203, no. 9, pp. 1–5, 2021.

[8] E. Ishita, S. Fukuda, Y. Tomiura, and D. W. Oard, "Using text classification to improve annotation quality by improving annotator consistency," *Proc. Assoc. Inf. Sci. Technol.*, vol. 57, no. 1, 2020, doi: 10.1002/pra2.301.

[9] A. Alsaeedi, "A survey of term weighting schemes for text classification Abdullah Alsaeedi," vol. 12, no. 2, pp. 237–254, 2020.

[10] J. Chen, "Spam mail classification using back propagation neural networks," *Appl. Comput. Eng.*, vol. 5, pp. 438–449, Jun. 2023, doi: 10.54254/2755-2721/5/20230617.

[11] S. Kaddoura and S. Henno, "Dataset of Arabic spam and ham tweets," *Data Br.*, vol. 52, p. 109904, 2024, doi: https://doi.org/10.1016/j.dib.2023.109904.

[12] O. Fourkioti, S. Symeonidis, and A. Arampatzis, "Language models and fusion for authorship attribution," *Inf. Process. Manag.*, vol. 56, no. 6, p. 102061, 2019, doi: 10.1016/j.ipm.2019.102061.

[13] A. A. Aladeemy *et al.*, "Advancements and Challenges in Arabic Sentiment Analysis: A Decade of Methodologies, Applications, and Resource Development," *Heliyon*, p. e39786, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e39786.

[14] M. Ghiassi, S. Lee, and S. R. Gaikwad, "Sentiment analysis and spam filtering using the YAC2 clustering algorithm with transferability," *Comput. Ind. Eng.*, vol. 165, p. 107959, 2022, doi: https://doi.org/10.1016/j.cie.2022.107959.

[15] C. C. Abalorio, A. M. Sison, R. P. Medina, and G. A. Dalaorao, "Applying EMONO Variants to Multi-Class Sentiment Analysis for Short-Distance Inter-Class Frequency of Term," vol. 71, no. 4, pp. 1938–1947, 2022.

[16] Z. Cai *et al.*, "Multi-schema prompting powered token-feature woven attention network for short text classification," *Pattern Recognit.*, vol. 156, p. 110782, 2024, doi: https://doi.org/10.1016/j.patcog.2024.110782.

[17] M. Doğan, D. Aslan, V. Gürmeriç, A. Özgür, and M. Göksel Saraç, "Powder caking and cohesion behaviours of coffee powders as affected by roasting and particle sizes: Principal component analyses (PCA) for flow and bioactive properties," *Powder Technol.*, vol. 344, pp. 222–232, 2019, doi: https://doi.org/10.1016/j.powtec.2018.12.030.

[18] S. R. Guda, "The Repository at St . Cloud State Evaluation of Email Spam Detection Techniques," 2022.

[19] J. Zimmermann, L. E. Champagne, J. M. Dickens, and B. T. Hazen, "Approaches to improve preprocessing for Latent Dirichlet Allocation topic modeling," *Decis. Support Syst.*, vol. 185, p. 114310, 2024, doi: https://doi.org/10.1016/j.dss.2024.114310.

[20] Q. Ouyang, J. Tian, and J. Wei, "E-mail Spam Classification using KNN and Naive Bayes," *Highlights Sci. Eng. Technol.*, vol. 38, pp. 57–63, Mar. 2023, doi: 10.54097/hset.v38i.5699.

[21] C. M. Shaik, N. M. Penumaka, S. K. Abbireddy, V. Kumar, and S. S. Aravinth, "Bi-LSTM and Conventional Classifiers for Email Spam Filtering," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 2023, pp. 1350–1355. doi: 10.1109/ICAIS56108.2023.10073776.

[22] T. Dogan and A. K. Uysal, “A novel term weighting scheme for text classification: TF-MONO,” *J. Informetr.*, vol. 14, no. 4, p. 101076, 2020, doi: 10.1016/j.knosys.2012.06.005.

[23] J. Attieh and J. Tekli, “Supervised term-category feature weighting for improved text classification,” *Knowledge-Based Syst.*, vol. 261, p. 110215, 2023, doi: https://doi.org/10.1016/j.knosys.2022.110215.

[24] F. Shehzad, A. Rehman, K. Javed, K. A. Alnowibet, H. A. Babri, and H. T. Rauf, “Binned Term Count: An Alternative to Term Frequency for Text Categorization,” *Mathematics*, vol. 10, no. 21. 2022. doi: 10.3390/math10214124.

[25] Z. Tang, W. Li, and Y. Li, “An improved supervised term weighting scheme for text representation and classification,” *Expert Syst. Appl.*, vol. 189, p. 115985, 2022, doi: https://doi.org/10.1016/j.eswa.2021.115985.

[26] R. Palacharla and V. K. Vatsavayi, “A New Supervised Term Weight Measure Based Approach for Text Classification,” *Rev. d’Intelligence Artif.*, vol. 36, pp. 395–407, Jun. 2022, doi: 10.18280/ria.360307.

[27] M. Panwar, J. Jogi, M. Mankar, M. Alhassan, and S. Kulkarni, “Detection of Spam Email,” *Am. J. Innov. Sci. Eng.*, vol. 1, pp. 18–21, Dec. 2022, doi: 10.54536/ajise.v1i1.996.

[28] Y. Kontsewaya, E. Antonov, and A. Artamonov, *Evaluating the Effectiveness of Machine Learning Methods for Spam Detection*, vol. 190. 2021. doi: 10.1016/j.procs.2021.06.056.

[29] A. Sinha and S. Singh, “A DETAILED STUDY ON EMAIL SPAM FILTERING TECHNIQUES,” *Int. J. Data Sci. Anal.*, vol. 10, Oct. 2020.