T.C. AYDIN ADNAN MENDERES UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

CSE402 Graduation Thesis 2, Spring 2022
Supervisor: Asst Dr. Mohamed KURDI

Final Report

(Bachelor of Science Thesis)

By:
Mert Can BENLİ, Student ID: 171805059
Emre KUTLU, Student ID: 171805034

PLAGIARISM STATEMENT

This report was written by the group members and in our own words, except for quotations from published and unpublished sources which are clearly indicated and acknowledged as such. We are conscious that the incorporation of material from other works or a paraphrase of such material without acknowledgement will be treated as plagiarism according to the University Regulations. The source of any picture, graph, map or other illustration is also indicated, as is the source, published or unpublished, of any material not resulting from our own experimentation, observation or specimen collecting.

Project Group Members:

Name, Lastname	Student Number	Signature	Date
Emre KUTLU	171805034		
Mert Can BENLİ	171805059		

Project Supervisors:

Name, Lastname	Department	Signature	Date
Mohamed KURDİ	Computer Engineering		

ACKNOWLEDGEMENTS

We would like to express our greatest appreciation to our beloved teacher and supervisor Mohamed KURDI. We are thankful to our teacher for his ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project. A special acknowledgement goes to our classmate Erman DERİCİ who helped us in completing the project by exchanging interesting ideas, sharing his experience and collabrating in the implementation phase.

KEYWORDS

Feature Selection, Text Classification, Ant Colony Optimization, Feature Extraction.

ABSTRACT

Feature selection is the heart of any classification system. Feature selection is mainly used for reducing the high dimensionality of feature space in text categorization. To increase the performance of the text classifier, certain feature selection methods have been used. Over the last decade besides the traditional feature selection algorithms like Chi-Square, Information Gain, Document Frequency, meta-heuristic approaches and swarm optimization algorithms became popular. This paper firstly gives an overview about Ant Colony Optimization (ACO), then discuss the well known feature selection algorithms. Lastly, includes comparisons and evaluating which one is performing best.

ÖZET

Özellik seçimi, herhangi bir sınıflandırma sisteminin kalbidir. Özellik seçimi, esas olarak metin kategorizasyonunda özellik uzayının yüksek boyutluluğunu azaltmak için kullanılır. Metin sınıflandırıcının performansını artırmak için belirli öznitelik seçim yöntemleri kullanılmıştır. Son on yılda Chi-Kare, Bilgi Kazanımı, Belge Sıklığı gibi geleneksel öznitelik seçim algoritmalarının yanı sıra meta-sezgisel yaklaşımlar ve sürü optimizasyon algoritmaları popüler hale geldi. Bu makale öncelikle Karınca Kolonisi Optimizasyonu (ACO) hakkında genel bir bilgi vermekte, ardından iyi bilinen öznitelik seçim algoritmalarını tartışmaktadır. Son olarak, karşılaştırmaları ve hangisinin en iyi performans gösterdiğini değerlendirmeyi içerir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ACO	Ant Colony Optimization
AS	Ant System
TF	Term Frequency
IDF	Inverse Document Frequency
CV	Count Vectorizer
DF	Document Frequency
SVM	Support Vector Machine
NB	Naïve Bayes
DT	Decision Tree
RF	Random Forest
TSP	Traveling Salesman Problem

1. INTRODUCTION

1.1. Description of the Problem

Due to emerging web technologies and improvements on the computer hardware today's computers are processing gigabytes of data under a second and are creating millions of web pages and documents. Therefore, classifying billions of documents manually is not a good choice anymore. However, these improvements also made computers able to handle such humanly tasks like document-classification and text categorization.

1.2. Project Goal

The main goal of text classification is to assign a document to one or more classes or categories [1]. The document will be tagged with these predefined one or more classes. If there are two labels (classes), it is simply called binary classification. If three or more labels are predefined and the document only belongs to one of them, it is multiclass labeling. If three and more labels exists, and the document belongs to multiple of them, it is specified as a multi-label problem [2]. Nonetheless, pure textual data are meaningless to computers and computers cannot form relations and associations between document type and document categories. In order to extract features from raw textual data and make machine learning algorithms to perform classification we are using feature extraction and feature selection algorithms.

Feature extraction and feature selection are the heart of any classification system. One of the main problems in the feature selection is the size and high dimensionality of these features [3]. Performance of the classification algorithm directly dependent on these so-called selected features. This paper provides a new Ant Colony Optimization (ACO) based feature selection model and includes comparisons of formerly used feature selection methods.

1.3. Dataset

We are using Reuters dataset in our research [4]. The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd. and Carnegie Group, Inc.) in 1987.

The original dataset contains 21578 documents and their related topics. However, due to performance, hardware and elapsed time issues we only used 8490 documents from the dataset. These documents are labeled among 7 pre-defined categories.

1.4. Ant Colony Algorithm

Ant Colony algorithm firstly introduced in the nineties for solving optimization and traveling problems by Dorigo [5]. Natural ants find the most optimal paths by using feedback system via pheromones. Thus, other ants follow these pheromone trails and choose the most favorable path. After emerging it gained popularity and many researchers have developed their own ACO systems. Although ACO is generally used for optimization problems like Travelling Salesman Problem, researchers started to use ACO for various other fields like feature selection, classification etc.…

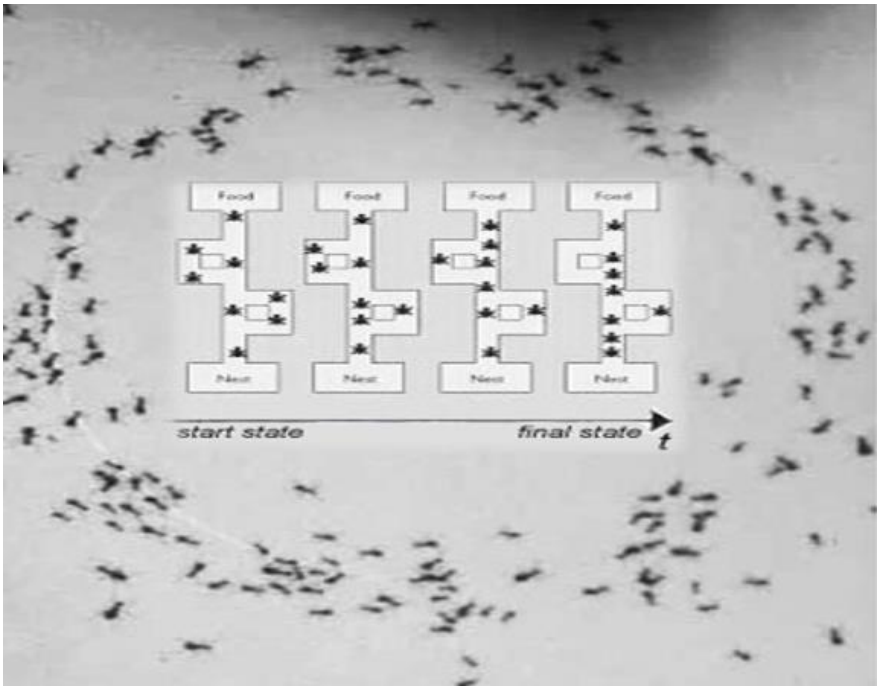

Figure 1: Example Iteration of Ant Colony

Figure 1 briefly shows and explains iterations of ACO. Notice that in the beginning ants are distributed to paths randomly and stochastically. After few iterations ants follow the pheromone paths to locate the food source. However, there are still few ants exploring the other directions despite pheromone favorable trails. This random behavior prevents colony to looping in the local optimum forever. First developed ACO algorithm is Ant System (AS) by Dorigo and colleagues [6]. Dorigo and his colleagues applied AS to well-known traveling salesman problem and have taken remarkable results. In the algorithm each ant travels through the graph.

1.5. Project Activities and Schedule

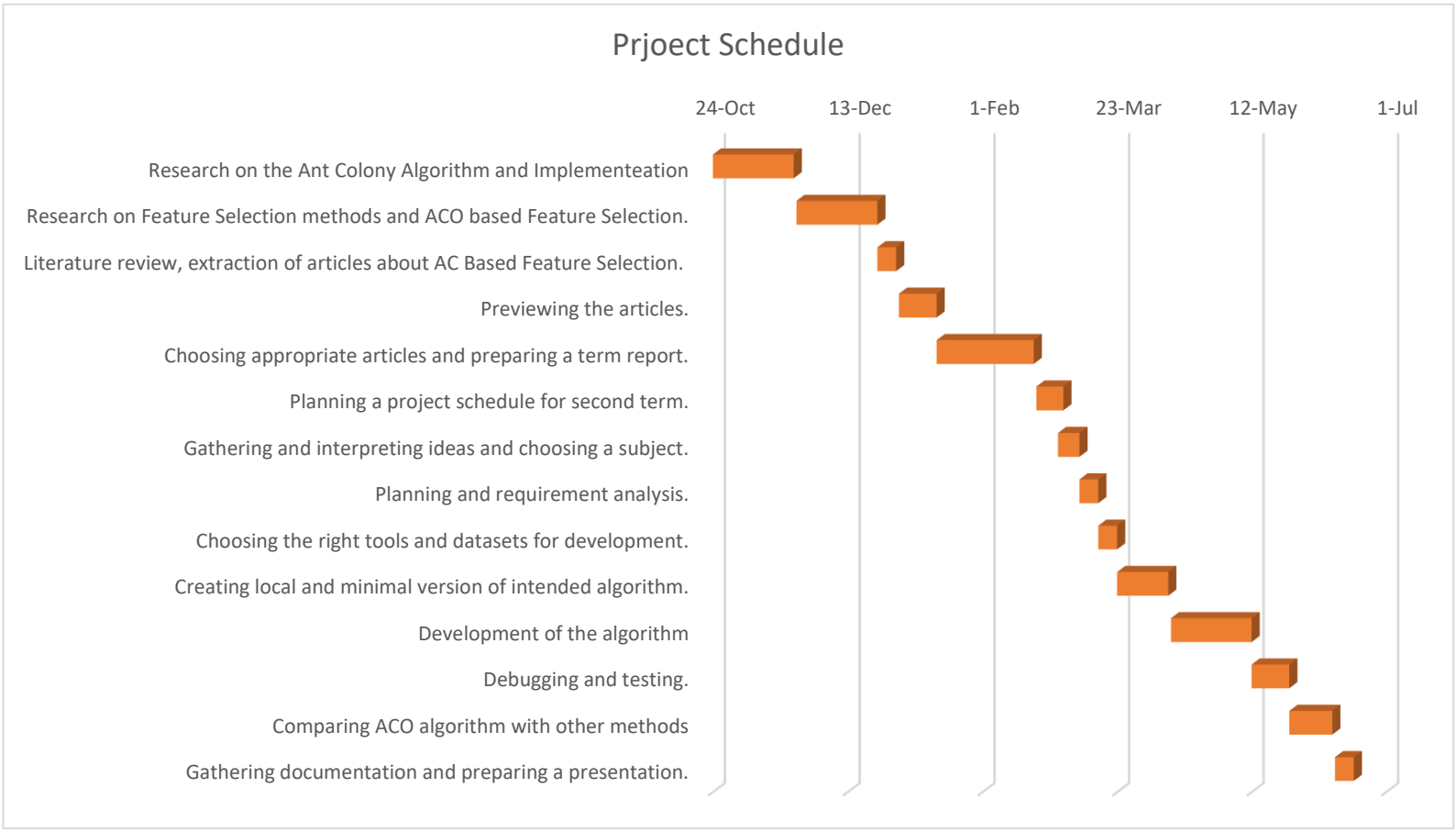

Figure 2: Schedule

1.6. Overview of Text Feature Selection Methods

As mentioned in the Section 1 the main problem of feature selection and feature extraction is high dimensionality of the vector space and feature subset. This dimensionality brings another problems like curse of dimensionality and overfitting models. According to [3] 109 of 175 publications use statistical feature selection approaches and 155 of 175 papers used Bag of Words (BoW) model. The most popular and trending approaches for measure relevance are Information Gain(IG), Document Frequency(DF), Mutual Information and Chi-Square (CHI). The sub-sections 1.6.1, 1.6.2 and 1.6.3 provides brief overview and discuss the advantages and disadvantages of these methods via recent publications.

1.6.1. Information Gain

Information gain is reduction of entropy [9] or 'surprise', unpredictability in simple words. A larger information gain means that lower entropy or 'surprise'. Entropy simply measures how much information there is in a random feature and its probability distribution.

$$Gain(t) = -\sum_{k=1}^{|C|} P(C_i) * logP(C_i) + P(t) * \sum_{k=1}^{|C|} P(C_i|t) * logP(C_i|t) + P(t) * \sum_{k=1}^{|C|} P(C_i|t) * logP(C_i|t)$$

Formula 1: Information Gain

Formula 1 states that information gain of term t is associated with occurrence probability of term t where c denotes category.

The weakness of information gain is giving more importance to Document Frequency (DF) rather than word frequency. For instance, word features called w1 and w2 are appearing in the same document. However, w1 is showing up much more frequently than w2. Despite the fact that w1 is much more common in the same document their impact for IG is same which is unhelpful [7]. Moreover [8] noted that one of the big flaws of IG is attaching different values to features that have similar or same impact. Even though IG is still one of the most popular [3] and widely used feature space reduction methods it has some flaws. Publications [7] and [8] reveals the disadvantages and improvements on the IG-based feature selection approaches on text classification.

1.6.2. Chi Square

Chi-square is a statistical test used to examine the differences between categorical variables from a random sample to judge goodness of fit between expected and observed results.

$$Chi - Square(t_k, c_i) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)}$$

Formula 2: Chi-Square

Formula 2 shows that how CHI method works. Where N equals to total number of documents, A equals to number of documents in the label, B equals to number of documents containing term, C equals to number of documents in the label c [9]. CHI method has shown remarkable results in text classification but is coming with some disadvantages. According to [9] when top twenty features selected the number of features per label has tendency to vary. 11 of top 20 features belong to one category and remaining 9 of them are in another category. Thus, classification of these features will predominant. The f-measures for given 2 category Sport and Business %93,7 and %87,6 respectively. Other classes scored poorly for this reason. There are other recent publications using CHI based feature selection for text classification. According to [2] Suzuki [10], Agnihotri et al [11], Bahassine et al[12], Sun et al[13] are some of the text classification studies based on Chi Square based feature selection. [14] states that Chi-Square (CHI) method performs better than IG. Accuracy values of CHI and IG for 2000 features are %95,03 and %92,50 respectively.

1.6.3. Document Frequency

Document Frequency is one of the simplest feature selection methods which selects features according to their 'Document Frequency'. In simple words Document Frequency (DF) works with the word counts. The higher the word's count in the same document higher its frequency. However, this simple method comes with a few disadvantages. For example, given words 'a, the, an'. These words occur in every single text in English at super-high frequencies. In order to DF to work well these irrelevant, uninformative words must be cleaned from the corpus. Study [15] compares DF, IG, and CHI methods using SVM classifier with Pang & Lee and 20-newsgroup dataset with 1000 features. Accuracy scores of DF, IG, and CHI methods are 0.6080, 0.6100, and 0.6030 respectively using Pang & Lee dataset. For 20-newsgroup results are 0.8684, 0.8861 and 0.9091 respectively. Under the same circumstances with same classifier CHI method have the best results.

2. RELATED WORKS, IMPLEMENTATION and TESTS

2.1. Related Works

Meta-heuristic approaches for feature selection have become very popular in the recent decade. Besides the traditional statistical based approaches like mentioned in the Section 3. Genetic algorithms, Swarm Optimization Algorithms and Hybrid approaches are very commonly used feature selection methods nowadays. According to our research one of the first multi-label classification studies using ACO is [13].

Agdham et al [3] developed ACO based feature selection system and ACO scored better than IG,CHI and GA.Overall accuracy of ACO for classifying 10 categories using Reuters-21578 dataset is %77.1343. IG, CHI and Genetic Algorithm (GA) scored %70.3791, %72.204 and %75.8898.

In another study Saraç et al. [14] tested ACO based feature selection on categorizing the web pages.They achieved better results than well-known traditional approaches IG and CHI using the WebKB [16] dataset .

Meena et al. [17] proposed an enhanced ACO based feature selection system which increased the macro average score from %84.21 to %86.35. They received best results with 150 iterations and creating 500 ants during each iteration. Also stated that algorithm gives the best results when equal importance assigned to heuristic information and pheromone deposits.

Mesleh et al. [18] using Support Vector Machine (SVM) developed ACO feature selection system for text classification. Using the best 160 features ACO scored better than IG, CHI and Mutual Information feature selection methods. Macro Precision values of aforementioned methods are %94.145(ACO), %91.64(CHI) and %85,11(IG). Not only ACO has better precision values its macro recall measures and f1 scores are better.

In another study [19] text classification system based on ACO is used. From 6217 features the best 200 feature subset is selected in the best case. Average accuracy for best 200 feature subset is %89.

Ahmad et al. [20] utilized ACO based feature selection for sentiment analysis. Using the pheromone decay parameter 0.8, relative weight of feature subset length as 0.2 and weight of classifier performance as 0.8 they received better scores than IG method using KNN as classifier. Average F-measure for ACO is %82.7 while IG scores %73.1.

Wajeed et al. [21] showed that the number of ants generated per iteration greatly impacts the performance of the classifier. In their case increasing number of ants from 5 to 10 improved classifier accuracy.

Alongside the pure ACO based or pure statistical based approaches [20] proposed IG-ACO hybrid feature selection method. However, they did not include evaluation and performance tests.

Imani et al. [22] are also used hybrid approach. Applying the CHI algorithm first and later using ACO they have taken improved results. Nonetheless their study shows Macro and Micro F1 scores of pure ACO based feature selection is achieved better results than IG, CHI and GA methods.

2.1. Preprocessing

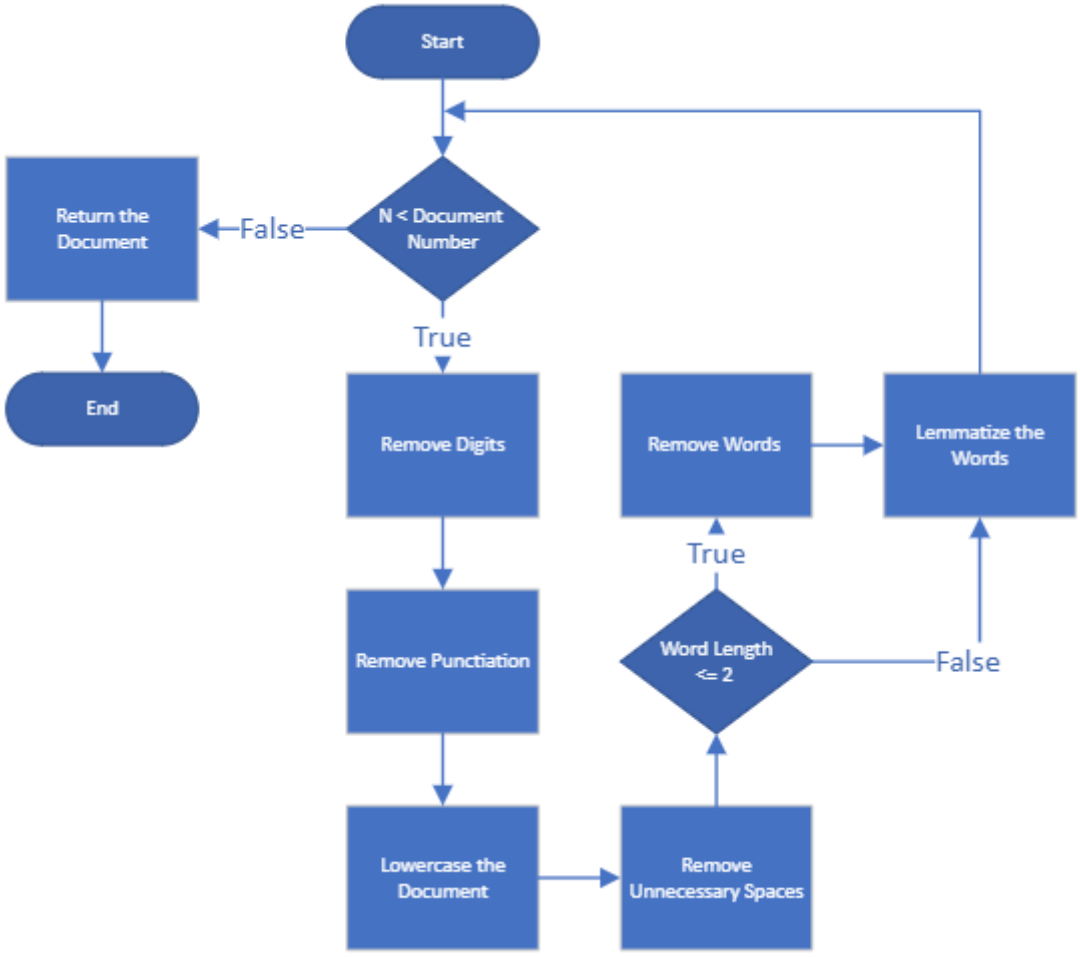

Figure 3: Preprocess

As we know machine learning needs data in the numeric form. We basically used encoding technique (TF-IDF, Word2Vec) to encode text into numeric vectors. But before encoding, we first need to clean the text data and this process to prepare (or clean) text data before encoding is called text preprocessing, this is the very first step to solve the NLP problems.

The textual data contains a lot of misleading information inherently. To make our data more reliable and classifier performance better, we have applied these steps shown in Figure 2. To explain briefly we :

- Removed digits, punctuation, unnecessary spaces, words which is shorter than 3 letters since they don't imply anything about the subject of the document
- Lowercased the document
- Lemmatized the words to understand the root of the words.

class	document
0	ASIAN EXPORTERS FEAR DAMAGE FROM U.S.-JAPA…
1	CHINA DAILY SAYS VERMIN EAT 7-12 PCT GRAIN STO…
2	JAPAN TO REVISE LONG-TERM ENERGY DEMAND D...
3	THAI TRADE DEFICIT WIDENS IN FIRST QUARTER\n ...
1	SRI LANKA GETS USDA APPROVAL FOR WHEAT PR…

Figure 4: Before Preprocess

class	document
0	asian exporter fear damage usjapan rift mount ...
1	china daily say vermin eat pct grain stock sur...
2	japan revise longterm energy demand downwards ...
3	thai trade deficit widen first quarter thailan...
1	sri lanka get usda approval wheat price food d...

Figure 5: After Preprocess

2.2. Feature Extraction

One of the major challenges is to choose the best possible numerical/vectoral representation of the text strings for running machine learning models. As far as we know, computers can only understand numerical data, while natural language data, for computers, are just text strings without any numerical or statistical information. So, we intend to bridge the gap by converting texts to numbers and conserving linguistic information for analysis. There are several ways to do so.

1. Count Vectorizers:

Count vectorizer (CV) is a way to convert a given set of strings into a frequency representation.

CV can clarify the type of text by the frequency of words in it, but its major disadvantages are:

- Its inability in identifying more important and less important words for analysis.
- It will just consider words that are abundant in a corpus as the most statistically significant word.
- It also doesn't identify the relationships between words such as linguistic similarity between words.

For CV we use the module within scikit-learn library.

(0, 187)	16
(0, 92)	12
(0, 146)	1
(0, 61)	8
(0, 35)	1

Figure 6: CV Vectors

CV calculates the term frequency of each word for each document and returns the values in the form of sparse matrix as shown in Figure 6. This may seem complicated but means 1. document contains 188. word 16 times. We can use this matrix directly in classifier to train our model. But only the frequency of a word in a document doesn't necessarily mean that word is impactful. Therefore, we need something more to understand the meaning of a word.

2.TF-IDF:

TF-IDF means Term Frequency - Inverse Document Frequency. This is a statistic that is based on the frequency of a word in the corpus, but it also provides a numerical representation of how important a word is for statistical analysis.

The formula for TF-IDF:

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ is\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

Formula 3: TF-IDF Equations

where d refers to a document, N is the total number of documents, df is the number of documents with term t.

The main reason why TF-IDF is better than CV is that it not only focuses on the frequency of words present in the corpus but also provides the importance of the words. We can then remove the words that are less important for analysis, hence making the model building less complex by reducing the input dimensions.

To sum up:

In TF-IDF we consider overall document weightage of a word. It helps us in dealing with most frequent words. TF-IDF weights the word counts by a measure of how often they appear in the documents. That's why it is the better choice in our situation.

To calculate TF-IDF we used TF-IDF vectorizer module within scikit-learn library.

(0, 57)	0.030275744
(0, 193)	0.035987635
(0, 53)	0.039368781
(0, 88)	0.035312115
(0, 64)	0.037402157

Figure 7: TF-IDF Vectors

These values represent the TF-IDF weights of each term in each document which are calculated by given Formula 3.

2.3.2. TF-IDF and CV for Training Model

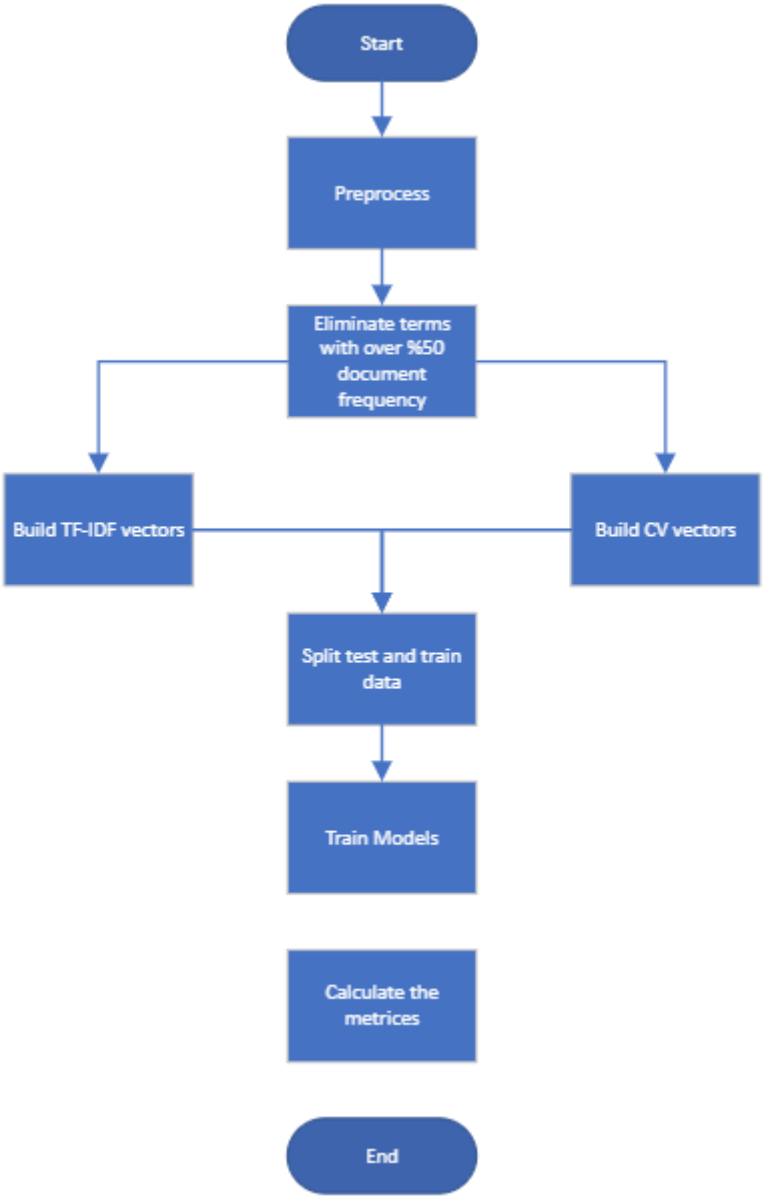

Figure 8: Model Training

To use machine learning algorithms, we need to preprocess our dataset so that these said algorithms can understand the data better. Then we calculated CV and TF-IDF vectors mentioned in Section 2.2. We split %30 of the data for testing purposes and remaining %70 for training. Our dataset contains 8,491 documents and their related labels. After splitting we had approximately 2,547 documents for testing and 5,944 for training. After stop word elimination and preprocessing the original dataset contains about 26,000 features however this is still too big. We decreased this number using built-in functions of CV and TF-IDF to 200.

Now we need to see how accurate our ML models are. There are few ways to understand how the model performs. We inspected three of these.

$$F1\ Score = \frac{Precision * Recall}{Precision + Recall}$$

Formula 4: F1 Score Equation

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Size\ of\ dataset}$$

Formula 5: Accuracy Equation

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Formula 6: Recall Equation

After these steps we proceeded to perform ML algorithms. Which are:

- SVM,
- NB,
- DT,
- RF.

	SVM	NB	DT	RF
F1 Score with CV	0.77576550924	0.68630581859	0.71420498713	0.79955795033
F1 Score with TF-IDF	0.87475991867	0.84678578631	0.78396391775	0.87793259105
Accuracy Score with CV	0.88383045525	0.83516483516	0.84065934065	0.89638932496
Accuracy Score with TF-IDF	0.93877551020	0.91601255886	0.87676609105	0.93445839874
Recall Score with CV	0.74647724510	0.61903408368	0.70965263435	0.78585663446
Recall Score with TF-IDF	0.86771188440	0.83389558573	0.77877959819	0.863930213529

Table 1: Results of Classifiers

The results without ACO feature selection are given in the Table 1. The TF-IDF method brings better results compared to CV. It is also noted that SVM with TF-IDF results are by far better than other classifiers.

True label	Predicted label
13	13
13	13
13	13
24	10
10	10
8	8
13	13
8	8
24	24
13	13
13	13
13	13
13	13
8	8
10	10
13	13
8	8
1	3
2	2
2	2

Figure 9: Predictions

Figure 9 represents the results of SVM classifier. As you can see its prediction percentage is convincingly high.

2.3. ACO based Feature Selection

This section includes our approach of ACO based feature selection. Our approach utilizes both ACO and document frequency feature selection methods. This hybrid approach uses IDF weights of each term to calculate their selection probabilities and pheromone values from the previous iterations. Firstly, we calculated each terms' IDF weights according to Formula 3 and stored them in a matrix. Then we initialized parameters for ACO. These parameters are:

- Epoch : Maximum iteration number
- Ant : Number of ants
- Feature count : Number of features to select
- $\tau(initial)$: Initial pheromone value
- α : Pheromone exponent value
- β : Heuristic exponent value
- ρ : Pheromone evaporation coefficient

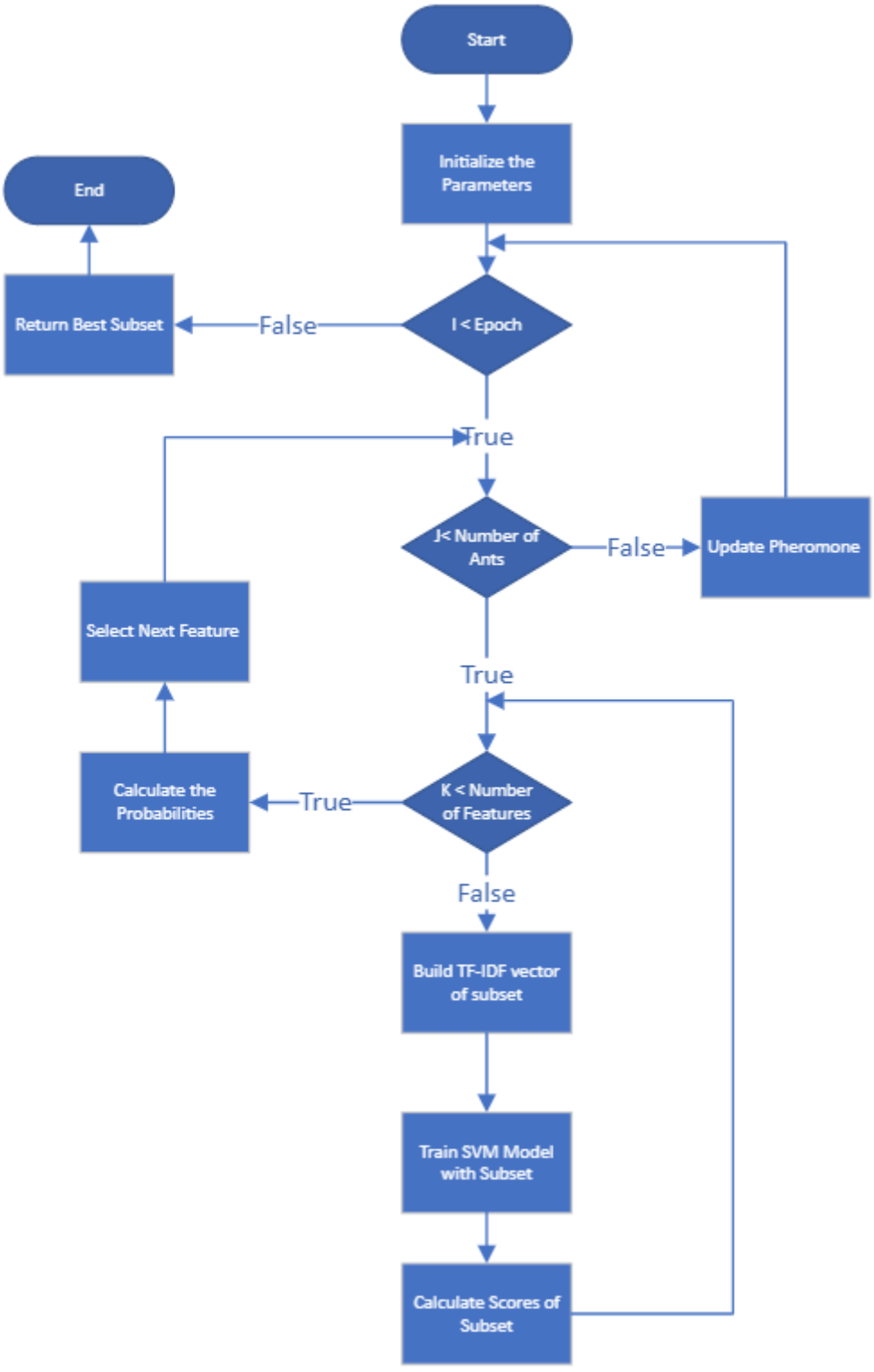

Figure 10: ACO

Figure 10 demonstrates our approach on ACO based feature selection algorithm. In our ACO, we used IDF weights for our heuristic therefore assigned features does not have same selection probabilities. The formula for our selection algorithm is:

$$Selection\ Probability = \frac{\tau(feature)^{\alpha} * IDF(feature)^{\beta}}{\sum_{0}^{n} \tau^{\alpha} * IDF^{\beta}}$$

Formula 7: Selection Probability

Selecting next feature is done by the Formula 5. The initial values we used are:

- α : 1
- β : 1
- τ : 1

Our transition formula defers from the other studies by heuristic method. Because we used TF-IDF weights as initial heuristic values so initial probability of each feature is different. After an ant builds its subset choosing N features, we train a model with TF-IDF vectors. With given SVM model, F1, Accuracy and Recall scores are calculated and stored in a dictionary. When each ant completes its path, pheromone values are updated according to their F1 measure scores.

Ant Precision Score :

Accuracy : 0.5074568288

F1 : 0.1599726986

Recall : 0.1767340912

Ant Path : 173, 111, 112, 65, 132

These results illustrate example run of a single ant with 5 features selected.

$$\tau(feature) = \rho\tau(feature) + Feature\ Pheromone(i)$$

Formula 8: Pheromone Update

Formula 6 shows pheromone update equation where τ(feature) represents pheromone of that particular feature. Initially all features have pheromone value of 1. ρ is evaporation rate which we decided and assigned the value of 0.4 after test runs.

$$Feature\ Pheromone(i) = \sum_{k=0}^{Ant} F1\ Score(k) * \tau(Initial)$$

Formula 9: Pheromone Update

This formula explains how each feature's pheromone values are calculated. Note that we are updating the pheromones using F1 scores therefore higher the F1 score, bigger the pheromone deposit.

2.4. Comparing With Other Feature Selection Algorithms

We have used Support-Vector Machine, Naïve Bayes, Decision Tree and Random Forest algorithms on Reuters dataset and result are given in table below:

Number of Ant	10	30	30
Iteration Count	10	10	20
Feature Count	100	100	100
Time(seconds)	134	431	908
F1	0.832	0.83	0.809
Accuracy	0.904	0.906	0.897
Recall	0.811	0.815	0.787

Table 2: SVM Values on Different Runs

Number of Ant	10	30	30
Iteration Count	10	10	20
Feature Count	100	100	100
Time(seconds)	17	64	113
F1	0.753	0.741	0.753
Accuracy	0.847	0.846	0.861
Recall	0.692	0.681	0.695

Table 3: NB Values on Different Runs

Number of Ant	10	30	30
Iteration Count	10	10	20
Feature Count	100	100	100
Time(seconds)	164	97	184
F1	0.835	0.742	0.766
Accuracy	0.906	0.847	0.853
Recall	0.82	0.736	0.756

Table 4: DT Values on Different Runs

Number of Ant	10	30	30
Iteration Count	10	10	20
Feature Count	100	100	100
Time(seconds)	36	503	1017
F1	0.755	0.828	0.856
Accuracy	0.857	0.903	0.917
Recall	0.751	0.81	0.839

Table 5: RF Values on Different Runs

Tables 2-5 are our experimental result for ACO based feature selection on different runs. 100 features selected by ACO performed as good as original 200 values. Therefore, we could say that the feature space is reduced to half and performed nearly the same. We tried several iteration runs, several ant numbers and deduced that after certain number of iterations performance is not improving. We found that the optimal number of ants is 20. In our experiments SVM and RF classifiers are the ones with the best results. However, their run time is significantly longer than NB and DT classifiers.

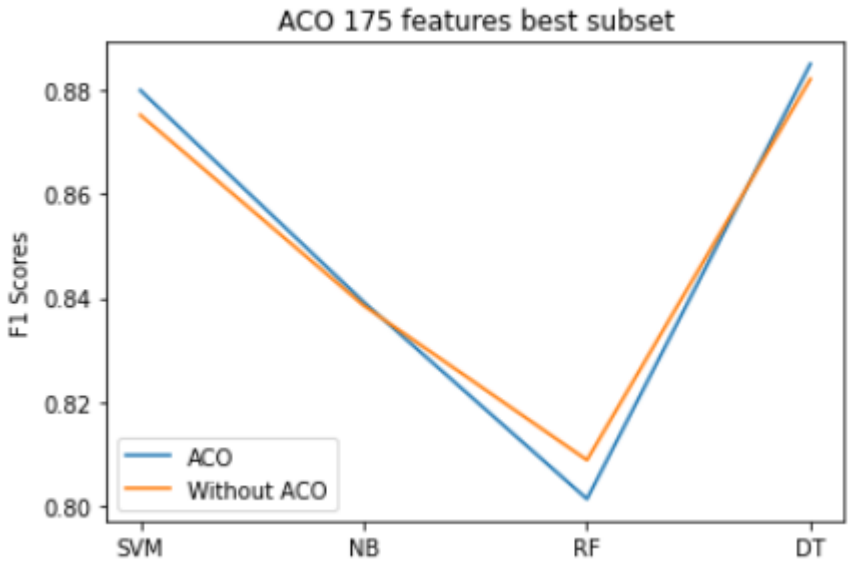

Figure 11: 175 Features selected by ACO

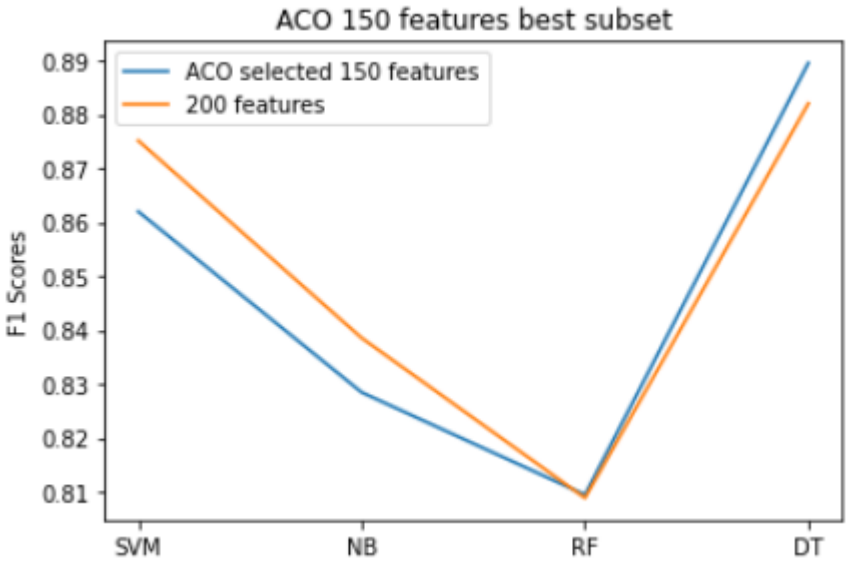

Figure 12: 150 Features selected by ACO

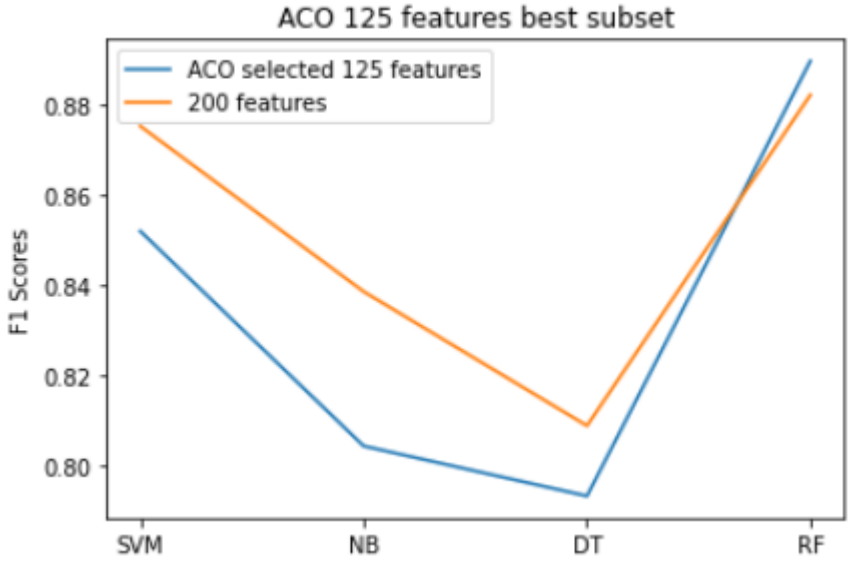

Figure 13: 125 Features selected by ACO

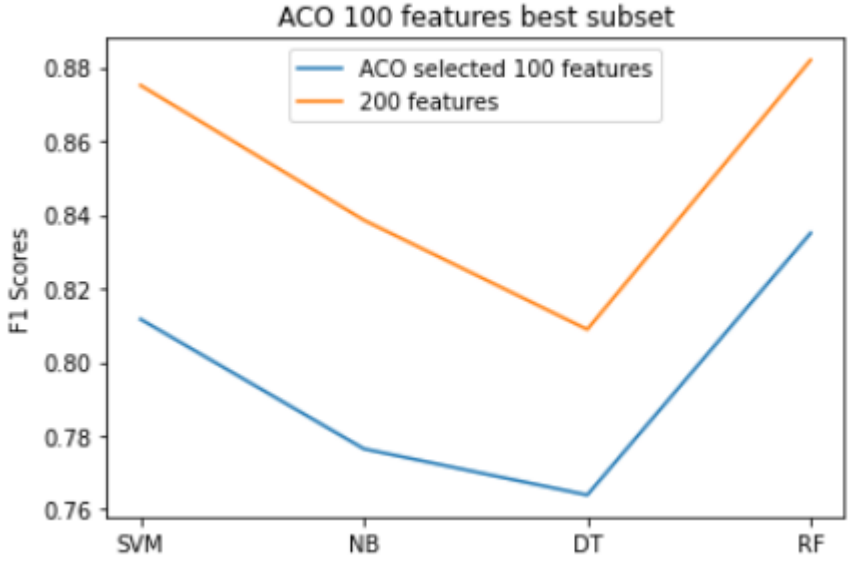

Figure 14: 100 Features selected by ACO

3. SUMMARY

In this work we have implemented ACO based Feature Selection for multilabel text-news classification problem. We have mainly used two feature extraction models which are Bow and TF-IDF. Before extraction step, we have preprocessed the Reuters Dataset then calculated IDF and DF values for each document for heuristic values. After these steps, we have initialized ants and made them build their sub-sets of features. The number of features to be selected for each ant is predefined. When an ant finishes its path, the selected path (features) is taken as parameters to train the model. The metrics (F1, Precision, Accuracy, Recall) have been calculated with each ant completing its path. According to these metrics, pheromone values of each feature get updated. These steps are repeated until the predefined iteration count is reached. The output of our project is the best selected sub-set.

For future work, word embeddings can be used as heuristic value.

3.1. Used Hardware and Software

All calculations, implementations and tests are done in Google Colab platform using their allocated machines. The specifications and hardware of used Google VM is:

- Python 3 Google Compute Engine Backend
- 12GB RAM
- 107 GB SSD Drive
- Intel® Xeon® CPU @ 2.20 GHZ 1 Core 2 Thread

The implementation is done in Python programming language version 3.10.5 which is latest version up to date. For the text preprocessing, stop word removal, lemmatizing and other purposes Python Library NLTK[23] and Scikit-learn[24] has been used.

References

[1] “Document Classification.” Wikipedia, Wikimedia Foundation, 08 January 2022, https://en.wikipedia.org/wiki/Document_classification.

[2] Pintas, J.T., Fernandes, L.A.F. & Garcia, A.C.B. Feature selection methods for text classification: a systematic literature review. Artif Intell Rev 54, 6149–6200 (2021). https://doi.org/10.1007/s10462-021-09970-6

[3] M. H. Aghdam, N. Ghasem-Aghaee and M. Ehsan Basiri, "Application of ant colony optimization for feature selection in text categorization," *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2867-2873, doi: 10.1109/CEC.2008.4631182.

[4] Reuters-21578 Text Categorization Collection Data Set
https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[5] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, Tech. Rep. 91-016, 1991

[6] M. Dorigo, V. Maniezzo, et A. Colorni, *Ant system: optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics--Part B , volume 26, number 1, pages 29-41, 1996.

[7] G. Wu and J. Xu, "Optimized Approach of Feature Selection Based on Information Gain," *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, 2015, pp. 157-161, doi: 10.1109/CSMA.2015.38.

[8] S. Rastogi, "Improving Classification Accuracy of Automated Text Classifiers," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 1-7, doi: 10.1109/ICRITO.2018.8748498.

[9] Said Bahassine, Abdellah Madani, Mohammed Al-Sarem, Mohamed Kissi, Feature selection using an improved Chi-square for Arabic text classification, Journal of King Saud University - Computer and Information Sciences, Volume 32, Issue 2, 2020, Pages 225-231, ISSN 1319-1578.

[10] Deepak Agnihotri, Kesari Verma, and Priyanka Tripathi. 2016. Computing Correlative Association of Terms for Automatic Classification of Text Documents. In Proceedings of the Third International Symposium on Computer Vision and the Internet (VisionNet'16).
DOI: https://dl.acm.org/doi/10.1145/2983402.2983424

[11] J. Sun, X. Zhang, D. Liao and V. Chang, "Efficient method for feature selection in text classification," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308201.

[12] S. Bahassine, A. Madani and M. Kissi, "An improved Chi-sqaure feature selection for Arabic text classification using decision tree," *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 2016, pp. 1-5, doi: 10.1109/SITA.2016.7772289.

[13] Allen Chan and Alex A. Freitas. 2006. A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In Proceedings of the 8th annual conference

on Genetic and evolutionary computation (GECCO '06). Association for Computing Machinery, New York, NY, USA, 27–34. DOI: https://doi.org/10.1145/1143997.1144002

[14] Esra Saraç, Selma Ayşe Özel, "An Ant Colony Optimization Based Feature Selection for Web Page Classification", *The Scientific World Journal*, vol. 2014, Article ID 649260, 16 pages, 2014.

[15] Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods, Applied Soft Computing, Volume 86, 2020, 105836, ISSN 1568-4946.

[16] http://www.cs.cmu.edu/~webkb/

[17] M. Janaki Meena, K.R. Chandran, A. Karthik, A. Vijay Samuel, An enhanced ACO algorithm to select features for text categorization and its parallelization, Expert Systems with Applications, Volume 39, Issue 5, 2012, Pages 5861-5871, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2011.11.081.

[18] A. M. Mesleh and G. Kanaan, "Support Vector Machine Text Classification System: Using Ant Colony Optimization Based Feature Subset Selection," 2008 International Conference on Computer Engineering & Systems, 2008, pp. 143-148, doi: 10.1109/ICCES.2008.4772984

[19] L. Jiao and L. Feng, "Text Classification Based on Ant Colony Optimization," *2010 Third International Conference on Information and Computing*, 2010, pp. 229-232, doi: 10.1109/ICIC.2010.242.

[20] Ahmad, Siti Rohaidah, Bakar, Azuraliza Abu, and Yaakub, Mohd Ridzwan. 'Ant Colony Optimization for Text Feature Selection in Sentiment Analysis'. 1 Jan. 2019 : 133 – 158.

[21] Luo, X. & Wang, Zhaoli & Lu, Yonghe. (2011). A study of text classification based on ant colony optimization. Libr. Inf. Serv.. 55. 103-106.

[22] Maryam Bahojb Imani, Mohammad Reza Keyvanpour & Reza Azmi (2013) A NOVEL EMBEDDED FEATURE SELECTION METHOD: A COMPARATIVE STUDY IN THE APPLICATION OF TEXT CATEGORIZATION, Applied Artificial Intelligence, 27:5, 408-427, DOI: 10.1080/08839514.2013.774211

[23] https://www.nltk.org

[24] https://scikit-learn.org/stable/