 http://dx.doi.org/10.14500/aro.12034

Enhanced Category-Feature Association Measure: A Robust Approach for Text Classification through Feature Selection

Soran S. Badawi[1], Ari M. Saeed[2†], Sara A. Ahmed[3] and Diyari A. Hassan[4]

[1]Language Center, Charmo University,
Chamchamal, Kurdistan Region – F.R. Iraq

[2]Department of Computer Science, University of Halabja,
Halabja, Kurdistan Region – F.R. Iraq

[3]Department of Computer Engineering, Komar University of Science and Technology,
Sulaimaniyah, Kurdistan Region – F.R. Iraq

[4]Department of Biomedical Engineering, Faculty of Engineering and Computer Science, Qaiwan International University,
Sulaimaniyah, Kurdistan Region – F.R. Iraq

***Abstract*—Text classification is one of the severe challenges for categorizing large and high-dimensional text data accurately and efficiently. Many features confuse the classification process, and feature selection (FS) strategies should be used to deal with the problem of high dimensionality. This paper proposes a novel FS technique based on enhanced category-feature association measure (ECFAM). ECFAM utilizes the existence and elimination of terms and the complicated relationships among the terms across different sections. This one-of-a-kind approach emphasizes the key role of ancillary terms in classifying and differentiating categories. The comparison is done on two important datasets, Reuters-21578 and 20-Newsgroups, through two widely employed supervised machine learning classifiers and one deep learning algorithm. Throughout our experiments, we investigate the feature sizes in nine different feature sets, ranging from 50 to 4000. Experimental data show that ECFAM always performs better than other methods concerning accuracy and computational cost.**

***Index Terms*— Dimension reduction, Feature selection, Long short-term memory, Multinomial Naive Bayes, Support vector machines, Text classification**

I. Introduction

In the modern digital era, classifying text data are considered one of the main tasks due to the massive availability of textual data. Historically, previous research in text classification (TC) was largely conducted on information retrieval and information science in different fields, including data mining, machine learning (ML), and pattern recognition (Palanivinayagam, et al., 2023).

TC obtains its essential functionality from feature selection (FS) because it reduces data dimensions while improving prediction models and making insights more interpretable. The need for efficient TC grows more crucial during this era of large multimedia data, which includes video, images, audio files, and text.

Applications such as social media analytics require TC. High-dimensional text data creates substantial processing problems with numerous obsolete features that reduce model accuracy. The identification and selection of the most essential attributes from textual data are achieved through FS techniques. FS reduces computational expenses and improves the classifier's ability to perform on new, unobserved datasets (Deng, 2019).

Recently, there has been a notable shift toward implementing ML algorithms and statistically based methods in TC. This involves categorizing documents into predefined groups using classification algorithms trained on labeled data, as manual processing is impractical and error-prone when handling massive digital datasets. Multiple factors, particularly ML algorithms, contribute to improving accuracy and efficiency (Erenel, Adegboye and Kusetogullari, 2020). Conducting research in the realm of TC requires three phases: Preprocessing, model selection, and classification. During preprocessing, textual data undergoes tokenization that includes splitting the sentence items into separate entities to prepare them to be numerically represented, eliminating the non-functional or insignificant words, normalization, and stemming, which entails the removal of prefixes and suffixes attached to a word. FS then identifies the most relevant features from the text data by examining tokens. However, the high-dimensional nature of text data increases the risk of overfitting, which can negatively impact classification

ARO-The Scientific Journal of Koya University
Vol. XIII, No. 2 (2025), Article ID: ARO.12034. 10 pages
DOI: 10.14500/aro.12034
Received: 02 February 2025; Accepted: 29 July 2025
Regular research paper; Published: 21 August 2025
†Corresponding author's e-mail: ari.said@uoh.edu.iq

performance (Mirończuk and Protasiewicz, 2018). To address this, FS is employed to eliminate unrelated or redundant features that lead to the reduction of dimensionality in the training data, thereby improving computational efficiency and classification accuracy (Gudakahriz, Moghadam and Mahmoudi, 2021; Zhou, Wang and Zhu, 2022).

FS techniques are generally categorized into filters, wrappers, and embedded methods (Pudjihartono, et al., 2022). This study prioritizes filter-based methods over wrappers and embedded techniques because filter methods operate independently of classification algorithms, making them more efficient and less computationally intensive. Filter-based methods streamline the FS process, enhancing the reliability and speed of TC.

In this paper, we introduce a novel FS technique and evaluate its impact on ML and deep learning algorithms, particularly in the context of the English language. Two datasets are utilized in this study (Adi and Celebi, 2014; Russell-Rose, Stevenson and Whitehead, 2002): The 20 NewsGroups dataset, which has approximately 20,000 news texts distributed across 20 distinct categories, and the REUTERS dataset, involving 11,228 news articles divided into 46 topics. We compared our proposed method against six other FS methods using support vector machines (SVMs), multinomial Naive Bayes (MNB), and long short-term memory (LSTM) classifiers to assess its effectiveness.

The remainder of the paper is designed as follows: Section 2 provides a literature overview of previous steered on filter-based FS. Section 3 provides information on the datasets used in this study, along with the proposed method. Section 4 evaluates and compares algorithm performance. The conclusion is crafted in the final section of the paper.

II. Related Works

The challenge of recognizing complex patterns in high-dimensional data has led to an increased focus on FS techniques in the domain of TC (Bhavani and Santhosh Kumar, 2021). Despite ongoing advancements, researchers have developed various FS methods, combining established filter-based approaches with feature transformation and wrapper-based techniques in innovative configurations. Examples of filter-based FS methods include document frequency (Dhal and Azad, 2022), information gain, mutual information (Omuya, Okeyo and Kimwele, 2021), enhanced Gini index (Miao, et al., 2022), and deviation from Poisson's distribution (López-González, et al., 2021). These methods have been extensively studied to enhance their effectiveness in TC applications.

TC accuracy suffers due to large textual data dimensions; thus, FS methods work as essential dimension reduction techniques in the TC domain. A research paper introduces the extensive feature selector (EFS), which applies corpus-based and class-based probabilities to solve TC feature problems. EFS goes through an evaluation process alongside nine established techniques for FS using MNB, SVM, and KNN classifiers. The new method, EFS, undergoes evaluation through testing on Reuters-21578, 20-Newsgroup, Mini 20-Newsgroup, and Polarity datasets using six different FS sizes. The experimental outcome reveals that EFS delivers superior results than other methods in multiple scenarios according to micro-F1 and macro-F1 score analyses (Parlak and Uysal, 2023).

Another research presents a new TC FS method using association analysis, which utilizes frequent and interrelated items for duplicate and irrelevant feature reduction instead of traditional measures such as distance, dependency, and consistency. This approach was implemented on the SMS spam dataset from the UCI repository. The proposed method accomplished a classification accuracy level of 95.155% while working with only 6% of the features, which proved effective in both feature redundancy reduction and classification improvement (Mamdouh Farghaly and Abd El-Hafeez, 2023).

Moreover, scholars attempted to combine filters and wrappers. For instance, Alyasiri, Cheah and Abasi, (2021) merged Information Gain (IG) with the Gray Wolf Optimizer (GWO) in a wrapper-based FS approach (Alyasiri, Cheah and Abasi, 2021). IG was utilized to find the top features, and GWO was implemented to refine the feature sets. This approach was tested on nine benchmark datasets using MNB. The combined approach has shown promising results compared to other FS algorithms.

Finally, a few studies introduced hybrid approaches, such as combining correlation-based filters with SVM and Recursive Feature Elimination, which was called SVM-RFE. This method created vigorous FS results by first identifying predominant and paired features through a correlation-based filters approach, then refining these into a concise feature subset using SVM-RFE, ultimately achieving high classification accuracy (Zhang, et al., 2014). Another study combines BERT for text embedding, Many-to-Many LSTM for token-level prediction, and Decision Templates for using outputs. This method is used for binary classification on IMDB movie reviews and multiclass classification on drug reviews. The research design demonstrates its proposed model, which outperforms current models based on evaluation metrics including accuracy, recall, precision, and F1-score (Jamshidi, et al., 2024)

Furthermore, another hybrid method binds a multi-scale CNN architecture with an LSTM model. The Multi-scale CNN extracts features from individual sentences, but the LSTM detects the patterns between words within their context. Both extracted feature vectors receive combined input into a SoftMax layer for classification purposes. This approach improved TC results on traditional CNN and LSTM models when specifically used for TC tasks (Dou, et al, 2023).

III. Filter-based Methods

FS filtering methods consist of picking related features due to their association with class labels. This approach is based on adding scores to each feature and selecting them by a pre-

 http://dx.doi.org/10.14500/aro.12034

defined threshold. Filters, which move unrelated variables from the picture before classifying, are performed by ranking methods (Parlak and Uysal, 2023). Of all the characteristics of a feature that brings value, the most important is the ability to give useful information about different data classes, which is also called feature relevance (Ige and Gan, 2024). Our FS process is the input to the classification algorithm. To measure the relationship between features, consider the distance value of features, the information about the features, the correlation, and consistency scores of the feature sets. Filters, in turn, give advantages such as speed, scalability, and the independence of learning algorithms, providing sequential classifier evaluation (Lyu, Feng and Sakurai, 2023). Nevertheless, filter methods generally return abstract results that do not highly differentiate classes as the interactive classifiers (Jain and Singh, 2018).

Filter methods can be divided into single-variable and multivariate types. Univariate filter methods, however, do not account for relationships between features, potentially leading to the selection of redundant features that negatively impact classification accuracy (Noroozi, et al., 2023). However, multivariate filter methods are independent of classifiers, which do not model feature interdependencies.

The TC application field uses diverse FS methods based on the filter. In this research, six filter methods are utilized: Among, CHI-Square (CH12), Comprehensively Measure FS (CMFS), Discriminative Features Selector (DFS), Distinguishing Feature Selector (DFS), Discriminative power measure (DPM), and Gini-index are the metrics used during FS. Table I presents the nomenclature of FS methods for TC.

A: Refers to the number of documents containing the term t in the positive class.
B: Refers to the number of documents containing the term t in the negative class.
C: Refers to the number of documents not containing the term t in the positive class.
D: Refers to the number of documents not containing the term t in the negative class.

The FS methods are different based on the formulas, as shown in Table II.

The selection of these six FS filter methods (CHI2, CMFS, DFS, Gini-index, DFSS, and DPM) was motivated by their diversity in theoretical foundations, encompassing statistical, information-theoretic, and projection-based techniques. Moreover, their relevance to categorical data is well-established, with CHI2 and Gini-index serving as widely adopted benchmarks, and their inclusion enables direct comparison against state-of-the-art approaches such as CMFS and DPM. Furthermore, these methods highlight limitations in existing techniques, including biases toward high-frequency features or the neglect of inter-feature dependencies, while ensuring reproducibility through publicly available implementations. This rigorous selection strategy ensures a comprehensive evaluation of the proposed method's robustness and advantages across diverse FS paradigms.

TABLE I
CONTINGENCY TABLE OF TERM T AND CLASS C

Terms/ Classes	c	$\bar{c}$
t	A	B
$\bar{t}$	C	D

IV. METHODOLOGY

A. Proposed Method

In this study, we designed a novel filter-based method called the ECFAM. The proposed approach employs sophisticated calculations to determine critical term-category relationships. By utilizing multiple conditional probabilities, it analyzes interconnections between terms and categories within given datasets.

Based on Table I, the equation of ECFAM for a term is as follows:

$$ECFAM = \left(\frac{\left(\frac{A}{A+C}\right)^2 \cdot \left(\frac{A}{A+B}\right)^2}{\left(\frac{B}{B+D} - \frac{C}{A+C}\right)^2 + 1} \right) \quad \text{eq (7)}$$

Equation 7 calculates the relationship and characteristics of the term based on its connection with the class. We calculate each term's discriminant power across all of the available classes by increasing the specific term's value that positively contributes to category definition. However, if a term is rarely present in other classes, its value will be decreased; this ultimately reduces the term's importance in all of the classes. The evaluation process consists of four key probabilities: $\left(\frac{A}{A+C}, \frac{A}{A+B}, \frac{C}{A+C} \text{ and } \frac{B}{B+D} \right)$. The first two probabilities $\left(\frac{A}{A+C} \text{and} \frac{A}{A+B} \right)$ provide information about the term's presence in the relevant class, while the third $\left(\frac{C}{A+C} \right)$ is meant to measure its absence in that class. The last probability $\left(\frac{B}{B+D} \right)$ function is to evaluate the significance of the term's presence in other classes. Through this comprehensive calculation process of the probabilities, we can select the relevant and unique terms that can boost the TC process.

The proposed algorithm assigns scores to terms by taking the terms' existence and relationship within specified classes into consideration, which can enhance the FS process for TC. If a term continuously appears in the documents of an assigned category, it receives a higher score to demonstrate its importance in that category. While a term that has lower or no existence in that category, a lower score is assigned to it. Moreover, if a term is common in other categories, its score should be reduced, as it is less distinctive in the selected category.

TABLE II
CONTINGENCY TABLE OF TERM T AND CLASS C

Feature-selection method	Formula
CH12: The CHI2 technique stands out as a significant approach to FS, assessing the deviation from the expected distribution when a feature's occurrence isn't tied to its class (Uysal and Gunal, 2012).	$CHI2 = \frac{N((A\cdot D)-(B\cdot C))^2}{(A+C)(B+D)(A+B)(C+D)}$ eq (1)
CMFS: The CMFS technique thoroughly assesses features, drawing from their class characteristics (Zhang, et al., 2014).	$CMFS = \frac{(A+C)}{N}.\frac{A}{A+C}.\frac{A}{A+B}$ eq (2)
DFS: Based on four predefined criteria, the DFS method ensures that FS aligns with specific conditions related to feature attributes and rejects non-informative attributes (Parlak and Uysal, 2023).	$DFS = \frac{\frac{A}{A+B}}{\left(\frac{A}{A+C}+\frac{A}{B+D}+1\right)}$ eq (3)
Gini-index: The Gini Index is a method that selects notable features based on their purity level (Saeed, et al., 2023). This method is suitable for different classification types.	$Gini-Index = \left(\frac{A}{A+C}\right)^2.\left(\frac{A}{A+B}\right)^2$ eq (4)
DFSS: The DFSS technique is based on a list of pre-determined criteria to streamline the process. The algorithm is crafted to select the most repeated features and rank them with higher occurrence rates while ignoring features present in every document. DFSS enhances class differentiation by focusing on informative FS (Yang, et al., 2012).	$DFSS = \frac{A}{A+C}.\frac{A}{A+C}.\left\lvert\frac{A}{A+B}.\frac{A}{A+C}\right\rvert$ eq (5)
The DPM serves as a computationally effective FS approach, considering both affirmative and bad attributes (Kim and Zzang, 2019). DPM's main objective is to discover the distinctive features that can be used for TC.	$DPM = \left\lvert\frac{A}{A+C}-\frac{B}{B+D}\right\rvert$ eq (6)

CH12: Chi-Square, CMFS: Comprehensively measure feature selector, DFS: Distinguishing feature selector, DFSS: Discriminative features selection, DPM: Discriminant projection method

The ECFAM is a robust tool that measures the strength of a term's association with a specific category, considering its connections to other categories. ECFAM uses squared terms and their interactions in its numerator for a detailed understanding of the term-category relationship. While the denominator accounts for the term's absence, ensuring a comprehensive analysis. This approach emphasizes the importance of terms in specific categories and highlights their unique ability to characterize and differentiate between them. ECFAM leads to a more informed and precise text feature set.

The proposed method is presented in pseudocode form below:

PROCEDURE:

1. Data preparation:
   a. Preprocess training data and generate document-term matrix DTM
   b. Compute term-document matrix DM = DTM^T
2. Build term-category matrix (TCM):
   FOR each document d in D_train:
   category c = class label of d
   FOR each term t in d:
   TCM[t][c]+ = 1
3. Compute modified feature score:
   FOR each term t in vocabulary:
   total_term_freq = sum(TCM[t])
   FOR each category c in C:
   //Calculate contingency table values with smoothing
   A = (TCM[t][c] + 1)/(total_term_freq + |C|)//Term t in category c
   B = (total_term_freq - TCM[t][c] - 1)/(term_freq-term_freq_per_cat[c] + |V|)//Term t, not in category c
   C = (term_freq_per_cat[c] - TCM[t][c] - 1)/(term_freq_per_cat[c] + |V|)//Not term t, in category c
   D = (term_freq - total_term_freq - term_freq_per_cat[c] + TCM[t][c] + 1)/(term_freq - term_freq_per_cat[c] + |V|)//Not term t, not in category c
   //Calculate modified score
   score[t][c] = $(A/(A+C))^{20}$* $(A/(A+B))^{20}$/(B/(B+D) - C/(A+C))^{20}+ 1
4. Feature ranking:
   FOR each term t:
   term_score[t] = max(score[t])
   sorted_features = sort terms by term_score in descending order and take top K
5. Feature balancing:
   Initialize cat_pos_neg[|C|][2] with zeros
   Initialize balanced_features as empty list
   FOR each term t in sorted_features:
   sign = sign of TCM[t][argmax(|TCM[t]|)]//Sign for category with max occurrence
   cat = argmax(|TCM[t]|)//Category with max occurrence
   IF balanced_features.size() < K:
   IF (cat_pos_neg[cat,1]/max(1, cat_pos_neg[cat,0]+cat_pos_neg[cat,1])) < NFR:
   IF sign > 0:
   cat_pos_neg[cat,0]+ = 1
   ELSE:
   cat_pos_neg[cat,1]+ = 1
   APPEND t to balanced_features
6. Model training and evaluation:
   a. Create feature matrices using subsets of sorted_features
   b. Train SVM classifiers on each subset

c. Evaluate accuracy on test set for each feature subset size

7. RETURN balanced_features and accuracy scores

B. Datasets

In this experiment, two distinct multiclass datasets were utilized. Firstly, the 20 News-Groups dataset was employed as a balanced dataset, comprising around 20k news articles spread across 20 different categories. The number of texts is balanced among classes and it is labeled for ten classes (Alt.atheism, Comp. graphics, Comp.os.ms-windows.misc, Comp.sys.ibm.pc.hardware, Comp.sys.mac.hardware, Comp. windows.x, Misc.forsale, Rec.autos, Rec.motorcycles, Rec. sport.baseball). Second, the REUTERS newswire dataset was employed as the imbalanced dataset, containing 11,228 newswires from Reuters. The number of classes is 10, which consists of (Earn, Acq, Money-fx, Grain, Crude, Trade, Interest, Ship, Wheat, Corn). All of these were labeled across 46 different topics (Adi and Celebi, 2014; Russell-Rose, Stevenson and Whitehead, 2002).

V. Classifiers

TC aims to categorize documents based on their assigned classes. On applying the algorithm, a novel model is utilized to determine the labels of the unseen data. In this study, MNB and SVM were utilized (Saeed, et al., 2022; Badawi, 2023).

A. MNB

The MNB algorithm is widely used in TC problems that rely on the term "t" probability fitting to class "c," as presented by Equation 8.

$$p(c,t) = \frac{p(c).p(t,c)}{p(t)} \quad \text{eq (8)}$$

The MNB estimates the chance of term t being in class c, with p(c) representing the odds of class c′s occurrence, p (t, c) denoting the likelihood of term t existing in class c, and p(t) indicating the prospect of term t within the dataset. The algorithm leverages the estimation of choosing the term in a particular class. (Abbas, et al., 2019).

B. SVM

The SVM transforms the training data into vectors. The feature of the vectors is divided into positive and negative classes. When the classifier is ready, new occurrences are taken for testing, to predict the categories.

$$(w_{TXi}+b \geq 1)(X_i \in C) \quad \text{eq (9)}$$

$$(w_{TXi}+b \leq -1)(X_i \in C') \quad \text{eq (10)}$$

where *w* signifies the weight of the vectors and b represents the bias value (Zong, et al., 2015).

C. LSTM

The LSTM employed in this study is a carefully designed sequential pipeline that fully benefits from the model's temporal modeling capabilities. Initially, we tokenize each document into its words and sequentially pass each token's embedding through the LSTM network, which are then converted into 300-dimensional embedding vectors using custom-trained Word2Vec representations. The LSTM processes these embeddings one at a time while maintaining its internal hidden state. This allows the model to not only capture the presence of sentiment-bearing words but also their temporal relationships and contextual dependencies.

This sequential processing approach enables the LSTM to model the nuanced ways in which sentiment and meaning evolve throughout the text. By maintaining and updating its internal representation as it processes each token, the LSTM can build increasingly sophisticated document-level representations that account for word order, negation, contrast, and other linguistic phenomena that are crucial for accurate TC.

To ensure domain-specific semantic representations optimized for our classification tasks, we trained custom Word2Vec embeddings directly from our experimental corpus. The Word2Vec model was trained using the Skip-gram architecture (sg=1) on the datasets, combining both training and test sets to maximize vocabulary coverage and semantic learning. We used 300-dimensional vector representations with a context window size of 10 tokens, which allows the model to capture both local and broader contextual relationships between words. We set the minimum word count threshold to 3, effectively filtering out extremely rare terms while retaining meaningful vocabulary. In addition, we utilized negative sampling with 10 negative samples per positive sample to optimize training efficiency.

The model was trained for 10 epochs, starting with an initial learning rate of 0.025 that linearly decayed to 0.0001, and we employed multi-core parallel processing to accelerate training.

D. Classifier Evaluation

To test the effectiveness of FS with TC techniques, F1 measure, and accuracy are considered as criteria for evaluating those methods' outcomes. Such algorithms are designed on two possible outcomes for each test-case instance, as shown in Table III, which gives an example with the label "actual" which shows the data before the application of the classifier, and "predicting" which indicates the data after the classifier has been applied to testing instances. In this context, TP, FP, FN, and TN represent true positive, false positive, false negative, and true negative classifications, individually. "True" indicates correctly classified cases, whereas "false" means misclassification. In addition, "positive" means terms that are detected in a particular class, but "negative" denotes those terms that are not connected to the class (Saeed, et al., 2022; Murshed, et al., 2022).

$$f1-measure = \frac{2.\ precision.recall}{precision + recall} \quad \text{eq (11)}$$

$$precision = \frac{TP}{TP + FP} \quad \text{eq (12)}$$

$$recall = \frac{TP}{TP + FN} \quad \text{eq (13)}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{eq (14)}$$

VI. Experiments and Results

The experiment of this study is implemented using ten-fold cross-validation techniques with MNB, SVM, and LSTM algorithms. Six FS methods, including the proposed one, are executed on two datasets. For each dataset, nine subsets of features are selected with sizes of 50, 100, 250, 500, 750, 1000, 2000, 3000, and 4000. The F1, accuracy, and time consumption of each TC algorithm are evaluated as shown in Tables IV–XIII. The highest score of each feature subset is bolded within the tables, while the highest score in each table is underlined.

Examining Table IV–XI, it is clear that the first column shows the total features, while the rest of the columns reveal the FS method used for comparison. The bold values are the superior values for each subset of features based on the feature selection methods, while the underlined bold ones are the superior values in each table. The results indicate that as the number of features increases, the F1 measure improves for nearly all feature sets.

As shown in Tables IV–VII, the accuracy scores achieved by different FS methods, such as DPM, CHI2, CMFS, DFS, GINI, DFSS, and ECFAM, across different subsets of features which are arranged from 50 to 4000 for 20 Newsgroup and Reuter datasets using SVM and MNB as two algorithms in TC. In Tables IV and V, when MNB and SVM on the 20 Newsgroup dataset are implemented, the ECFAM acquires the highest score for each (50, 100, 250, 500, 750, and 2000) subset of features while DFS exhibits the highest score for the subset of features (3000 and 4000), and GINI scores the highest value in 1000 subset of features. In Table VI, when MNB on the Reuter dataset is implemented, the ECFAM attains the highest score for each (50, 100, 250, and 500) while DPM obtains the highest score for each (750, 1000, 3000, and 4000) subset of features and GINI achieves the highest score in (2000) subset of features. In contrast, in Table VII, when SVM on the Reuter dataset is employed, the ECFAM receives the highest score for each (100, 500, 1000, 2000, 3000, and 4000) subset of features while DPM attains the maximum score for each subset of features (50, 250, and 750). Moreover, in Tables VIII and IX, when the 20 Newsgroup dataset is considered, the ECFAM scores remain at peak for each (50, 500, 750, and 2000) when MNB is implemented while SVM is implemented, the ECFAM is the highest method for each subset of features (50, 250, 500, and 750). However, when the subset of features is (100, 1000, 3000, and 4000), each GINI and DFSS obtain the notable score, and GINI is highest with (250) subset of features in Table VIII. In addition, in Table IX, when the feature subset is (1000), CMFS exhibits the highest score, while GINI displays the highest score for the rest of the features. Moreover, in Tables X and XI, when the Reuter dataset is considered, the ECFAM shows the highest score for each

TABLE III
Confusion Matrix

Prediction/Actual	Positive	Negative
Positive	TP	FP
Negative	FN	TN

TABLE IV
MNB - Accuracy of 20-NEWs Group

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	28.00	21.97	26.55	15.51	31.31	25.69	**32.34**
100	33.51	30.66	35.25	25.19	38.17	37.73	**38.52**
250	41.44	41.40	47.62	40.60	50.39	49.19	**50.90**
500	48.94	47.17	53.41	51.13	54.97	54.86	**56.19**
750	53.33	50.58	56.60	53.28	58.13	56.35	**57.61**
1000	54.94	53.15	55.68	54.20	**59.48**	58.02	59.43
2000	59.55	56.33	57.87	59.17	59.23	59.77	**62.12**
3000	59.73	58.83	59.47	**61.40**	60.75	59.77	61.09
4000	60.63	60.41	59.85	**62.94**	61.38	60.71	61.71

MNB: Multinomial Naive Bayes, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure

TABLE V
SVM Accuracy-score of 20-NEWs Group

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	26.47	22.90	24.50	17.55	30.21	22.28	**33.38**
100	31.20	34.65	32.45	23.32	34.67	30.24	**37.49**
250	37.75	42.41	38.78	34.87	40.46	38.47	**43.12**
500	42.86	46.23	41.25	40.63	47.84	42.93	**50.62**
750	44.68	49.55	44.67	44.83	55.29	53.28	**56.29**
1000	45.17	52.12	47.35	49.06	56.20	55.19	**58.04**
2000	46.61	54.14	50.65	53.19	58.74	57.34	**62.08**
3000	51.43	56.80	53.28	60.44	60.12	59.86	**61.30**
4000	53.48	57.10	56.52	61.45	61.24	60.21	**61.59**

MNB: Multinomial Naive Bayes, SVM: Support vector machine, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure

Table VI
MNB Accuracy-score of Reuter

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	70.74	65.11	62.42	57.45	69.12	67.50	**71.35**
100	78.71	68.58	66.62	65.85	75.36	74.05	**80.44**
250	84.98	79.82	77.09	78.44	83.90	80.90	**85.29**
500	86.37	83.90	81.83	84.37	86.29	83.60	**86.45**
750	**87.10**	84.60	82.83	85.87	86.37	85.14	86.45
1000	**86.98**	85.33	83.79	86.87	86.95	86.02	86.25
2000	88.06	86.41	86.02	87.37	87.83	86.87	85.41
3000	**88.33**	86.87	87.10	87.64	88.22	87.18	84.06
4000	**88.26**	87.68	87.41	87.72	88.22	87.37	83.13

MNB: Multinomial Naive Bayes, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure

http://dx.doi.org/10.14500/aro.12034

TABLE VII
SVM Accuracy-score of Reuter

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	**76.55**	71.43	70.20	65.96	75.24	72.70	74.70
100	82.25	76.63	74.36	72.04	80.98	78.63	**82.56**
250	**86.29**	79.94	80.05	81.29	85.02	83.21	85.75
500	**86.83**	83.25	82.63	84.98	86.64	85.02	**86.83**
750	**86.91**	83.33	83.06	86.02	86.79	85.44	86.87
1000	86.68	84.33	84.14	86.48	86.56	86.14	**86.87**
2000	86.41	86.06	85.60	86.25	86.02	86.06	**86.91**
3000	85.91	86.25	85.52	85.91	86.02	85.71	**86.83**
4000	85.75	86.02	85.44	85.60	85.52	85.48	**86.79**

SVM: Support vector machine, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure

TABLE VIII
MNB F1-score of 20-NEWs group

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	28.65	23.59	29.99	15.57	36.57	29.47	**37.22**
100	34.34	34.45	40.33	28.65	**43.39**	42.46	43.02
250	41.72	43.76	50.41	45.84	52.95	**53.59**	53.47
500	49.28	47.33	54.21	54.96	55.69	56.13	**57.02**
750	53.57	49.97	56.88	56.52	58.47	56.85	**57.92**
1000	55.07	52.08	54.46	55.48	**59.65**	58.27	59.58
2000	59.69	54.88	56.59	58.99	57.85	59.92	**62.16**
3000	58.33	57.37	58.04	**60.49**	59.26	59.31	59.62
4000	59.18	58.83	58.43	**61.57**	59.93	59.96	60.19

MNB: Multinomial Naive Bayes, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure
The bold values are the superior values for each subset of features based on the feature selection methods, while the underlined bold ones are the superior values in each table

TABLE IX
SVM F1-score of 20-NEWs group

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	29.03	21.06	30.52	18.54	37.00	31.87	**38.36**
100	33.44	39.77	39.99	31.16	**42.90**	42.29	42.47
250	39.62	41.80	46.02	43.75	48.08	47.38	**48.69**
500	45.45	43.18	46.78	46.81	48.69	48.31	**49.45**
750	47.54	45.21	47.99	44.61	49.32	48.64	**49.86**
1000	48.24	48.36	**50.50**	46.19	50.00	49.13	50.14
2000	50.11	50.63	51.65	47.10	**54.70**	50.06	51.36
3000	51.73	52.73	52.73	48.15	**58.70**	47.18	52.73
4000	52.83	52.78	53.80	49.31	**59.80**	47.17	53.76

SVM: Support vector machine, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure

TABLE X
MNB F1-score of Reuter

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	71.38	60.71	63.00	50.19	69.92	68.46	**71.42**
100	79.43	66.33	67.86	62.56	76.55	75.28	**80.90**
250	85.16	79.83	77.97	78.83	84.10	81.53	**85.57**
500	**86.45**	83.90	82.18	84.28	86.34	83.85	86.20
750	**87.07**	84.55	83.05	85.74	86.45	85.24	85.99
1000	**86.99**	85.19	83.83	86.69	86.99	86.12	85.63
2000	**87.95**	86.05	85.60	86.91	87.56	86.50	84.27
3000	**87.97**	86.40	86.60	87.07	87.86	86.66	82.55
4000	87.72	87.17	86.66	87.06	**87.77**	86.70	81.28

MNB: Multinomial Naive Bayes, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure

TABLE XI
SVM F1-score of Reuter

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	71.69	67.99	64.37	57.35	70.03	67.63	**70.52**
100	78.90	74.60	69.61	65.76	77.28	74.43	**79.94**
250	**84.25**	78.14	75.92	77.33	82.36	79.98	83.81
500	84.82	81.71	79.18	82.49	84.65	82.22	**85.00**
750	**85.07**	81.76	79.84	83.79	84.91	83.17	85.04
1000	84.89	82.70	81.37	84.67	84.80	84.30	**85.03**
2000	84.74	84.23	83.60	84.54	84.23	84.27	**85.08**
3000	84.09	84.40	83.64	84.15	84.23	83.90	**85.00**
4000	83.88	84.11	83.62	83.80	83.64	83.66	**84.95**

SVM: Support vector machine, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure
The bold values are the superior values for each subset of features based on the feature selection methods, while the underlined bold ones are the superior values in each table

TABLE XII
T-test Comparison of EFCAM with Other Method

Compared method	t-statistic	p-value
DPM	4.9436	0.0011
CHI2	6.3448	0.0002
CMFS	6.2709	0.0002
DFS	3.1036	0.0146
GINI	2.0797	0.0711
DFSS	3.2888	0.0110
LSTM	5.7352	0.0004

SVM: Support vector machine, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, LSTM: Long short-term memory

(50, 100, and 250) when MNB is implemented. In contrast, when SVM is implemented, the ECFAM demonstrates the highest value for each subset of features (50, 100, 500, 1000, 2000, 3000, and 4000). Conversely, when the subset of features is (500, 1000, 2000, 3000, and 4000), DPM stays at the top, and GINI scores extraordinary with 4000 subsets of features in Table X. In addition, in Table XI, when the feature subset is (250, 750), DPM achieves the highest score.

Fig. 1 illustrates the frequency of successful FS method outcomes in each table. Each cell's score represents the count of feature subsets (columns) corresponding to specific types of FS methods (rows). These scores indicate the success or failure of FS methods within each table, as depicted below:

As shown in Fig. 1, the horizontal bar represents the six FS methods with the developed one, while the vertical bar is the number of tables, which is eight. Each bar is the number of times FS methods displayed the best scores in each table. ECFAM remains almost successful for each feature subset (50, 100, 250, 500, 750, 1000, 2000, 3000, and 4000). Moreover, the ECFAM success measure is 87.5% for the (50, 100, and 500) subset of features, while for the (250 and 750) subset of features is 75%, and for the (1000, 2000, 3000, and 4000) subset of features is 25%. While CHI2, DFSS is getting the

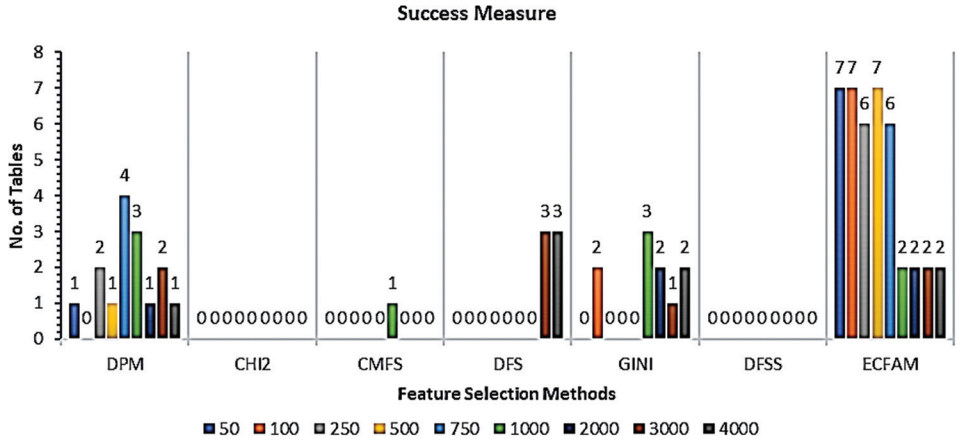

Fig. 1. Number of times features selection methods succeeded in each table.

TABLE XIII
TIME CONSUMPTION

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	0.033	0.135	0.036	0.037	0.032	0.034	0.031
100	0.045	0.142	0.033	0.046	0.031	0.035	0.040
250	0.046	0.151	0.037	0.038	0.041	0.039	0.043
500	0.044	0.174	0.039	0.039	0.043	0.042	0.039
750	0.045	0.157	0.043	0.039	0.044	0.043	0.041
1000	0.048	0.159	0.045	0.046	0.045	0.045	0.042
2000	0.052	0.173	0.046	0.053	0.047	0.046	0.044
3000	0.054	0.208	0.048	0.055	0.048	0.052	0.047
4000	0.058	0.261	0.053	0.056	0.051	0.055	**0.049**

DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure.

TABLE XIV
LSTM -ACCURACY SCORE OF 20 NEWSGROUP

No. features	DPM	CHI2	CMFS	DFS	GINI	DFSS	ECFAM
50	23.97	22.73	18.43	14.06	22.24	21.28	**24.22**
100	28.10	29.97	25.33	17.57	26.23	**28.49**	26.96
250	33.88	35.04	31.79	21.63	32.72	34.78	**36.62**
500	35.20	36.96	38.59	22.11	36.64	39.74	**41.67**
750	41.33	39.37	40.78	24.35	39.37	40.34	**43.18**
1000	44.46	41.27	47.72	30.50	43.04	46.42	**49.18**
2000	47.90	44.52	50.80	34.73	44.17	48.71	**51.73**
3000	51.30	48.47	54.68	39.69	47.79	54.84	**53.77**
4000	52.56	53.61	57.31	42.96	54.86	56.23	**58.53**

LSTM: Long short-term memory, DPM: Discriminant projection method, CH12: Chi-Square, CMFS: Comprehensively measure features selector, DFS: Discriminative features selector, DFSS: Discriminative features selection, ECFAM: Enhanced category-feature association measure.

worst score, which is 0. CMFS succeeds only once when the number of feature subsets is 1000, while DFS only two times succeeds when the number of feature subsets is (3000, 4000). GINI is getting zero scores for (50, 250, 500, and 750) feature subsets and DMP for (100) feature subsets. Additionally, DMP received the lowest score for 100 feature subsets, while it achieved the highest score for 750 feature subsets. It maintained the same score for 50, 500, 2000, and 4000 feature subsets.

Following this, we examined the statistical significance of the proposed method's performance improvements over existing FS techniques. We performed paired t-tests on classification accuracy scores across nine feature sizes, as shown in Table XII.

The proposed method demonstrated statistically significant improvements when compared to DPM ($p = 0.0011$), CHI2 ($p = 0.0002$), CMFS ($p = 0.0002$), DFS ($p = 0.0146$), and DFSS ($p = 0.0110$), with all p-values being less than 0.05. However, the difference with GINI was not statistically significant ($p = 0.0711$). These results support the effectiveness of ECFAM in delivering consistent and superior performance.

In an attempt to discover the computational cost of ECFAM, we measured the time required to process the data. We employed an MNB classifier on the Reuters dataset as a baseline to measure the computational cost, as shown in Table XIII.

CHI2 consistently requires the most processing time across all feature counts, ranging from 0.13 s for 50 features to 0.26 s for 4,000 features. In contrast, the other methods mainly stay below 0.05 s. On the other hand, ECFAM demonstrates excellent computational efficiency, requiring only 0.03 s for 50 features and maintaining a runtime of just 0.04 s even with 4,000 features. This showcases remarkable scalability, as ECFAM is the only method that does not exceed 0.05 s at the highest feature count. This demonstrates ECFAM's robustness in handling increased dimensionality. Overall, ECFAM is well-suited for applications requiring quick processing and stable accuracy with large feature sets, where memory constraints are less of a concern. Its ability to maintain performance while keeping processing time nearly constant is a key advantage for large-scale ML.

To evaluate ECFAM's performance with state-of-the-art deep learning techniques, we used LSTM as a classifier and employed the same FS methods to compare with ECFAM. This approach enables us to determine how effectively ECFAM's FS capabilities translate to modern deep learning architectures, which is essential for understanding its practical utility in contemporary ML applications. The implementation of LSTM provides a more rigorous testing ground for assessing the quality of the selected features, as deep learning models can capitalize on more complex feature interactions that simpler classifiers might overlook. Table XIV shows the results.

ECFAM shows steady improvement as feature count increases, with particularly significant performance jumps between 50 and 500 features (24.22–41.67) and more gradual gains thereafter. This pattern indicates that ECFAM effectively captures important discriminative features early in the selection process while continuing to identify useful features at higher counts. The method's single underperformance occurs at 3000 features, where DFSS marginally outperforms it (54.84 vs. 53.77), though ECFAM reclaims its leading position at 4000 features.

VII. Conclusion

This paper introduces ECFAM, a novel FS method, as a pioneering approach for ranking features in TC. It selects the best-ranked features based on their scores. Unlike conventional methods, ECFAM considers the relationships between terms across different categories rather than merely their presence or absence. In addition to ECFAM, we have conducted comparative analyses with six other FS techniques within this domain. Our investigation involved using two distinct datasets in conjunction with three classifiers. The first discovery underscores the pivotal role of terms in defining specific categories and their potential to distinguish between them effectively. Conversely, the second finding suggests that no single method can universally excel across all feature dimensions. Nevertheless, ECFAM has consistently surpassed other FS techniques, consistently achieving the highest scores across various datasets. Our results unequivocally establish ECFAM's superiority, as it consistently outperforms all other FS methods and scenarios across most datasets.

VIII. Limitations and Future Work

While ECFAM demonstrates superior performance, certain limitations warrant consideration. The method's effectiveness may be impacted by highly imbalanced datasets, and its computational efficiency could be challenged in extremely high-dimensional feature spaces. In addition, the current formulation assumes term independence within categories, which may not fully capture linguistic dependencies in all languages. Future research directions include extending ECFAM's framework for multi-label classification scenarios, optimizing its computational complexity for real-time applications, and validating its performance across diverse multilingual datasets. These enhancements would further strengthen ECFAM's applicability while addressing current constraints.

References

Abbas, M., Ali Memon, K., Jamali, A.A., Memon, S., and Ahmed, A., 2019. Multinomial naive Bayes classification model for sentiment analysis. *IJCSNS International Journal of Computer Science and Network Security*, 19(3), p.62.

Adi, A.O., and Celebi, E., 2014. Classification of 20 news group with Naive Bayes classifier. In: *2014 22nd Signal Processing and Communications Applications Conference (SIU)*. IEEE, United States, pp.2150-2153.

Alyasiri, O.M., Cheah, Y.N., and Abasi, A.K., 2021. Hybrid filter-wrapper text feature selection technique for text classification. In: *2021 International Conference on Communication and Information Technology (ICICT)*. IEEE, United States, pp.80-86.

Badawi, S.S., 2023. Using multilingual bidirectional encoder representations from transformers on medical corpus for Kurdish text classification. *ARO-the Scientific Journal of Koya University,* 11(1), pp.10-15.

Bhavani, A., and Santhosh Kumar, B., 2021. A review of state art of text classification algorithms. In: *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, United States, pp.1484-1490.

Deng, X., Li, Y., Weng, J., Zhang, J., 2019. Feature selection for text classification: A review. *Multimedia Tools and Application*, 78, pp.3797-3816.

Dhal, P., and Azad, C., 2022. A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence,* 52(4), pp.4543-4581.

Dou, G., Zhao, K., Guo, M., and Mou, J., 2023. Memristor-based LSTM network for text classification. *Fractals*, 31(06), p.2340040.

Erenel, Z., Adegboye, O.R., and Kusetogullari, H., 2020. A new feature selection scheme for emotion recognition from text. *Applied Sciences*, 10(15), p.5351.

Ige, O.P., and Gan, K.H., 2024. Ensemble filter-wrapper text feature selection methods for text classification. *CMES-Computer Modeling in Engineering and Sciences*, 141(2), pp.1847-1865.

Gudakahriz, S.J., Moghadam, A.M.E., and Mahmoudi, F., 2021. Opinion texts clustering using manifold learning based on sentiment and semantics analysis. *Scientific Programming*, 2021, p.7842631.

Jain, D., and Singh, V., 2018. Feature selection and classification systems for chronic disease prediction: A review. *Egyptian Informatics Journal,* 19(3), pp.179-189.

Jamshidi, S., Mohammadi, M., Bagheri, S., Najafabadi, H.E., Rezvanian, A., Gheisari, M., Ghaderzadeh, M., Shahabi, A.S., and Wu, Z. 2024. Effective text classification using BERT, MTM LSTM, and DT. *Data and Knowledge Engineering*, 151, p.102306.

Kim, K., and Zzang, S.Y., 2019. Trigonometric comparison measure: A feature selection method for text categorization. *Data and Knowledge Engineering,* 119, pp.1-21.

López-González, J.L., Franco-Villafañe, J.A., Méndez-Sánchez, R.A., Zavala-Vivar, G., Flores-Olmedo, E., Arreola-Lucas, A., and Báez, G., 2021. Deviations from poisson statistics in the spectra of free rectangular thin plates. *Physical Review E,* 103(4), p.043004.

Lyu, Y., Feng, Y., and Sakurai, K., 2023. A survey on feature selection techniques based on filtering methods for cyber attack detection. *Information*, 14(3), p.191.

Mamdouh Farghaly, H., and Abd El-Hafeez, T., 2023. A high-quality feature selection method based on frequent and correlated items for text classification. *Soft Computing*, 27(16), pp.11259-11274.

Miao, Y., Wang, J., Zhang, B., and Li, H., 2022. Practical framework of gini index in the application of machinery fault feature extraction. *Mechanical Systems and Signal Processing,* 165, p.108333.

Mirończuk, M.M., and Protasiewicz, J., 2018. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications,* 106, pp.36-54.

Murshed, B.A.H., Abawajy, J., Mallappa, S., Saif, M.A.N., and Al-Ariki, H.D.A., 2022. DEA-RNN: A hybrid deep learning approach for cyberbullying detection in twitter social media platform. *IEEE Access,* 10, pp.25857-258571.

Noroozi, Z., Orooji, A., and Erfannia, L., 2023. Analyzing the impact of feature selection methods on machine learning algorithms for heart disease prediction. *Scientific Reports*, 13(1), p.22588.

Omuya, E.O., Okeyo, G.O., and Kimwele, M.W., 2021. Feature selection for classification using principal component analysis and information gain. *Expert Systems with Applications,* 174, p.114765.

Palanivinayagam, A., El-Bayeh, C.Z., and Damaševičius, R., 2023. Twenty years of machine-learning-based text classification: A systematic review. *Algorithms*, 16(5), p.236.

Parlak, B., and Uysal, A.K., 2023. A novel filter feature selection method for text classification: Extensive feature selector. *Journal of Information Science,* 49(1), pp.59-78.

Pudjihartono, N., Fadason, T., Kempa-Liehr, A.W., and O'Sullivan, J.M., 2022. A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics,* 2, p.927312.

Russell-Rose, T., Stevenson, M., and Whitehead, M. 2002. *The Reuters Corpus Volume 1-from Yesterday's News to Tomorrow's Language Resources*. European Language Resources Association (ELRA), Las Palmas.

Saeed, A.M., Badawi, S., Ahmed, S.A., and Hassan, D.A., 2023. Comparison of feature selection methods in Kurdish text classification. *Iran Journal of Computer Science*, 7, pp.55-64.

Saeed, A.M., Ismael, A.N., Rasul, D.L., Majeed, R.S., and Rashid, T.A., 2022. *Hate Speech Detection in Social Media for the Kurdish Language*. Springer, Cham, pp.253-260.

Uysal, A.K., and Gunal, S., 2012. A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems,* 36, pp.226-235.

Zhang, J., Hu, X., Li, P., He, W., Zhang, Y., and Li, H., 2014. A hybrid feature selection approach by correlation-based filters and SVM-RFE. In: *2014 22nd International Conference on Pattern Recognition*. IEEE, United States, pp.3684-3689.

Zhou, H., Wang, X., and Zhu, R., 2022. Feature selection based on mutual information with correlation coefficient. *Applied Intelligence,* 52(5), pp.5457-5474.