[1]Hardikkumar Harishbhai Maheta

[2] Chauhan Pareshbhai Mansangbhai

[3]Chintan Makwana

An Improved Feature Removal Approach for Classification of High Dimensional Feature Dataset

Abstract: - Extracting and evaluating pertinent information in a high-dimensional feature set is extremely difficult when dealing with a high-dimensional feature space. Classification methods require additional training time to generate a classification model. Every feature is not equally relevant in a high-dimensional feature collection. As a result, feature selection is a productive method for identifying vital features and eliminating unnecessary ones. Feature selection serves as a method of pre-processing before classification. It decreases the dimension of the dataset to shorten the training period required to develop a classifier. This research study aims to propose a novel feature subset selection method that establishes the relative importance of each feature using several criteria. The proposed approach ranks available features from high to low using a variety of feature-ranking approaches. Different feature ranking algorithms perform differently on the same dataset. It is challenging to obtain robust performance with just one feature ranking algorithm. To overcome this problem, we have used the Schulze rank aggregation method. The Schulze method combines multiple feature ranking techniques to assign a rank to each feature inside the dataset. This study presents an optimization strategy for heuristic search based on the backward feature removal method. It eliminates features according to the rank determined by the Schulze rank aggregation technique. In this paper, we evaluated the performance of the proposed method against the current state-of-the-art feature ranking techniques for high-dimensional feature set classification.

Keywords: Feature Ranking, Feature Subset Selection, Backward Feature Elimination, Hybrid Feature Selection, Schulze Method

I. INTRODUCTION

High-dimensional feature sets require more time to generate a classification model. All the features from the high-dimensional feature set do not take part in the development process of a classifier. Removing those unnecessary features before the creation of the classifier reduces the time to create the classifier [1]. Feature selection is a preprocessing phase in classification that reduces the dimension of the dataset. An ideal subset of features increases the classifier's performance from the perspective of evaluation metrics like classification accuracy and time complexity [2]. Utilizing an optimal subset of features in real-world applications can lead to simple and faster classification models. It also increases the understanding of classification rules described by the classification model. There are two primary components to feature selection procedures. (i) The production of various subsets of features using various search approaches and (ii) the Usefulness of various feature subsets obtained by search approaches using evaluation criteria [2–4]. There are three broad methods of feature selection approaches based on the interaction with the classifier. These three methods are (i) the Filter approach, (ii) the Wrapper approach, and (iii) the Embedded approach [3]. The filter approach uses an intrinsic characteristic of the data and provides a ranking of features as output. It generates results without interaction with the classification technique [3, 5]. It is the fastest approach among all approaches. The wrapper approach is associated with the specified classifier [3, 6]. It selects the best subset of features from the feature set using different search methods [5, 6]. There are three main search strategies available: (1) Sequential search, (2) Complete search, and (3) Random search [7, 8]. The evaluation metrics and validation criteria guide the choice of the best possible subset of features in the wrapper approach. Wrapper strategies perform better than filter strategies. However, it takes more time to discover an optimal subset of features because one must run a classification algorithm during the feature subset generation process [6]. The Embedded approach uses a mixture of the filter and wrapper approaches to complete feature selection. The Embedded approaches provide an excellent balance between classifier generation time and classification performance [3]. Feature selection aims to

[1]* Corresponding author: Assistant Professor, Department of Information Technology, Shantilal Shah Engineering College, Bhavnagar, Gujarat, India, Email: phd.hardikmaheta@gmail.com

[23]Assistant Professor, Department of Information Technology, Shantilal Shah Engineering College, Bhavnagar, Gujarat, India

discover meaningful information from a pool of data. It is beneficial in application domains that contain many features and minimal instances. Text mining and bioinformatics are two examples of such fields [4]. Researchers always choose a stable feature selection method in a real-world application, where slight changes in the dataset do not affect the efficacy of the entire feature selection process. Surprisingly, the robustness (stability) of the feature selection methods received a relative lack of attention in the literature [7, 9]. The stability metrics employed for the feature selection process have been the primary focus of recent research in this field. Different stability measures for feature selection are dependency, consistency, and information theory [9]. Kalousis et al. provide a comparative assessment of the stability of feature selection across multiple high-dimensional datasets [5, 9]. We emphasize the robust feature ranking approaches and optimal backward feature elimination strategies. In this research paper, We studied whether combining different feature ranking algorithms results in a more stable feature rank and a more stable classification performance. The Schulze method is effective for a combination of various feature ranking strategies. It gives a final rank list of features that provides a more stable performance than an individual rank list of different feature rank strategies [10].

The rest of the paper is structured as follows. We explain the research topic with the substantial development made thus far in section II. We described various rank aggregation methods used to generate rank for features in section III. We discussed a heuristic feature selection technique called optimized backward feature elimination in section IV. Section V presents the experimental findings of the suggested method on high-dimensional feature sets. Section VI brings this paper to a conclusion.

II. Problem Description

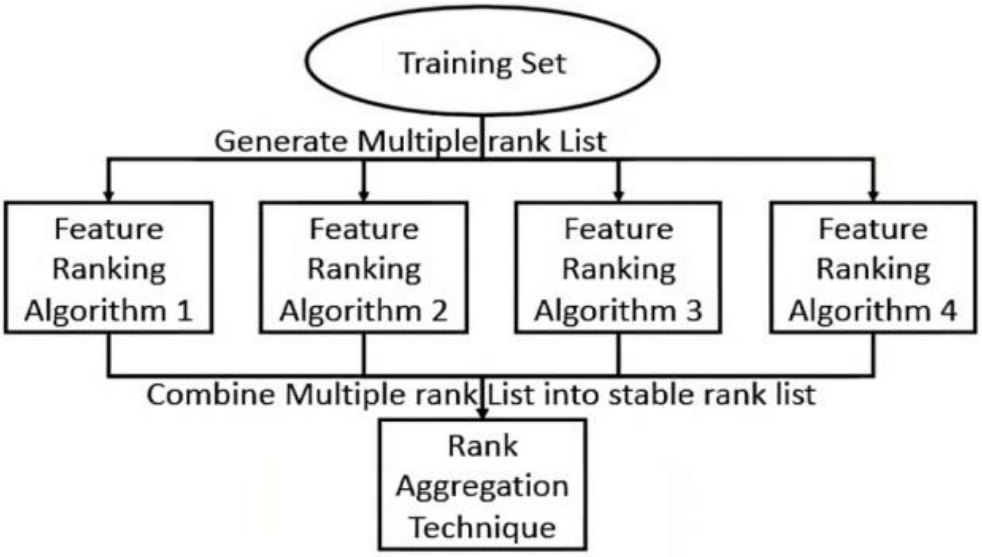
Wherever In many real-life applications, we have very high-dimensional feature sets. The feature selection technique applies to the high-dimensional feature set to select the most desirable features. We have an m-dimensional data collection that serves as the input for the algorithm for choosing features. The data set contains n number of data samples. A Matrix ($Data_{n*m}$) represents a dataset, where m represents the overall feature count in the data set, while n represents the total number of samples. Suppose $X=\{X(i) \mid i = 1, 2, .., m\}$ is an original feature set with m-dimension. Then, the purpose of the feature choice method is to find a new feature vector $Y=\{ Y (i) \mid i = 1, 2, 3, .., p\}$, which is a subset of the initial feature set X. It means $Y \subset X$ and $p \leq m$.

The classification algorithm takes the optimal subset of features. An ensemble learner produces predictive models. They compete with other single learners and provide better performance. This assertion is correct in the field of bioinformatics [11]. The current research examined used the ensemble notion to choose features [12]. The ensemble idea offers advantages over feature selection based on a single rank list, including a more stable feature list and superior classification accuracy [12]. However, the choice of how to aggregate the results is a part of the ensemble feature selection technique. In the following part, we discuss several possible rank aggregating methods. Many parts of the ranking process are the subject of rank aggregation strategies, such as giving the highest-ranked features greater weight or merging two or more established feature selection methods to improve performance [11, 13]. The prior research study demonstrates that selecting different rank aggregation strategies improves the classifier performance [13]. If we apply feature ranking procedure $R(D, F_{target})$ to dataset D, it outputs a list of features $F=\{F_1, F_2, ..., F_n\}$ ordered by decreasing importance $Imp(f_i)$ concerning F_{target}. The function $Imp(f_i)$ and the list of ranked features are distinct for each ranking method. If we use only one ranking technique, it may not give the best results for all datasets. A specialist in the field would favor a stable algorithm over an unstable one. We consider the issue of how reliable the ordering of features in a ranked list is, assuming we are unaware of the ranking technique. Ensemble feature ranking refers to this problem.

III. Rank Aggregation

The rank aggregation method aims to generate one final rank of the attributes using different available rank lists. Several ranking strategies, including Information Gain, Symmetric Uncertainty, Gain Ratio, OneR, Relief, Chi-square, etc., are available for the Feature selection technique. Each of the above-listed ranking techniques ranks the data set features. Rank states the importance of each attribute of the data set. Any specific ranking technique cannot provide the optimal rank for every feature of the feature set. If we combine rank from multiple ranking techniques, we can get advantages of all ranking techniques and also get a stable rank list [14]. We face two significant obstacles when we combine many feature rank lists into a single rank list. (i) How can several rank lists be generated? (ii) The aggregation function employed in rank aggregation. There are two different ways to generate rank lists. (i) The first one is to use the same dataset but apply multiple ranking techniques with

different criteria to generate different rank lists shown in Figure 1. (ii) The second one is to divide the data set into partitions and apply different ranking techniques on different segments of the datasets to generate multiple rank lists, as shown in Figure 2. In our experiments, we have used the first rank generation technique. The second issue is which type of aggregation function to use. There are primarily two types of rank aggregation: order-based rank aggregation and score-based rank aggregation. An individual rank list contains scores for each feature, and the accumulation process uses these scores to generate the final rank list for all attributes in score-based rank aggregation. For creating the final rank list, an order-based rank aggregation incorporates the order of each feature in each rank list. In this paper, we have focused on the order-based rank aggregation technique. In this research, We evaluated the features that made up the data set using six well-liked feature ranking algorithms. These techniques are Information Gain, Symmetric uncertainly, Gain Ratio, OneR, Relief, and Chi-Squared [15]. Borda(BD), Condorcet(CD), Schulze(SSD), and Markov Chain are some of the popular rank aggregation methods [2, 16, 17]. We have used the Schulze rank aggregation method to generate a stable rank list from multiple rank lists generated with the above-listed feature ranking techniques.



. Fig. 1 Use of Different Feature Ranking Algorithm

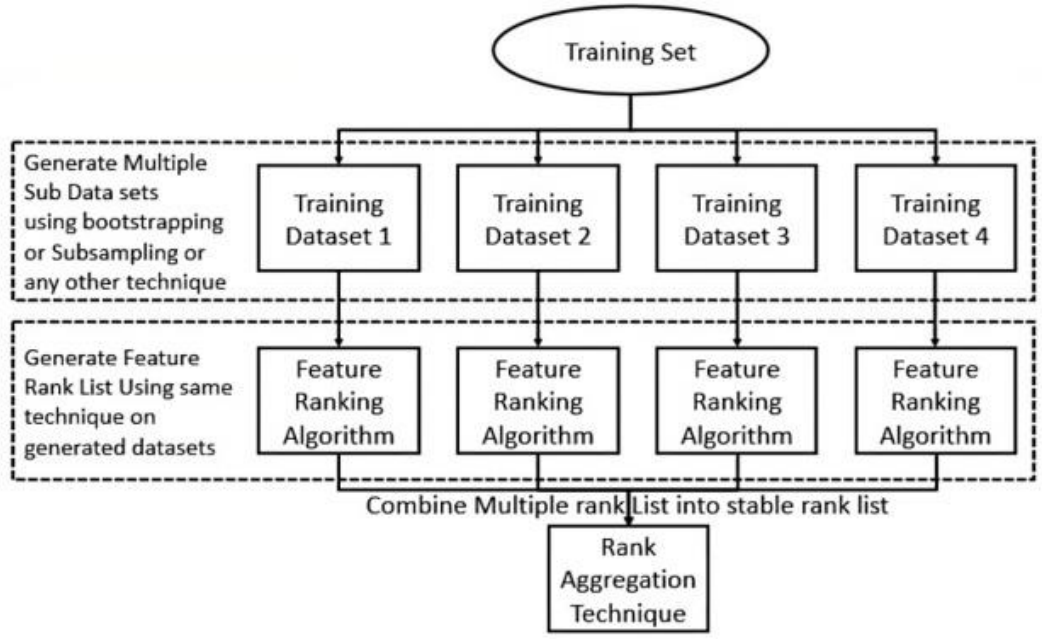


Fig. 2 Rank Aggregation

A. Feature Ranking Techniques

1) Information Gain (IG): A feature's Information Gain is its ability to predict the class of training data instances based on uncertainty [18]. According to Claude Shannon, an attribute with a higher uncertainty value provides valuable information, whereas a feature with a lower uncertainty value is irrelevant. To put it another way, if a coin has a head on both sides, then the outcome of tossing it does not produce any information. However, if a conventional coin has a head and a tail, then the result of tossing a coin provides information. The Information Gain (entropy) about feature Y following observation of feature X is the same as the Information Gain (entropy) about feature X following analysis of feature Y. The IG criterion has the drawback of favoring features with various values even when those features are not more informative than other features. It is possible to identify the features that are most relevant to the target class through information gain. IG is provided by, The steps are as follows:

$$IG=H(Y)-H(Y \mid X)=H(X)-H(X \mid Y) \quad (1)$$

a) Find the entropy of the target feature.

$$\text{Info}\,(D) = H(D) = -\sum_{i=1}^{n} p_i \log_2 (p_i) \quad (2)$$

where, P_i is the nonzero probability that a specific tuple in D belongs to class C_i.

b) For each feature, find the anticipated information needed To categorize a tuple from the target feature based on partitioning by that feature.

$$\text{Info}_{\text{feature}\ i}\,(D) = \sum_{j=1}^{m} \frac{|D_j|}{D} \times \text{Info}\,(D_j) \quad (3)$$

c) Now, Find Information Gain of each feature using the below formula:

$$\text{Gain}(\text{feature}_i) = \text{Info}(D) - \text{Info}_{\text{feature}\ i}(D) \quad (4)$$

d) Generate a rank list by ordering the Information Gain value of each feature in descending order.

2) Symmetrical Uncertainty (SU): Symmetrical uncertainty works on the drawback of Information Gain. It uses the entropy of individual features to overcome the problem of Information Gain. It normalizes its value between the range [0, 1] and rectifies Information Gain's bias towards features with enormous values [19].

$$SU(\text{feature}_i) = 2.0 \times \left[\frac{MI(X,Y)}{H(Y)+H(X)}\right] \quad (5)$$

where, MI(X,Y) = Mutual Information between Feature X and Feature Y, H(X) = Entropy of the target feature and H(Y) = Entropy of all features. The steps to find Symmetrical uncertainty are as follows:

a) Find the Entropy of the target feature and all features.

b) Find the Information Gain of every feature.

c) Following that, find the value of symmetrical uncertainty (SU) of every feature using the above listed equation (5).

d) Generate a rank list by ordering the value of symmetrical uncertainty of each feature in descending order.

3) OneR: OneR (One Rule) is a straightforward classifier that generates a decision tree with one level [2]. It can infer simple and accurate classification rules from data samples (instances). It can handle missing values of the dataset. It creates a Rule for each feature of the training data and then sets the Rule with the minimum error margin as its One Rule. The most common class for each feature determines the Rule of that feature. A Rule is a collection of feature values associated with their frequently associated class name. The Rule's error rate is the amount of training data instances in which the class of a feature value does not agree with the binding for that feature value in the Rule. The algorithm chooses the Rule randomly when two or more rules have the same error rate. In the nominal dataset, The OneR algorithm's steps are as follows:

a) Build the frequency table of each distinct value of every feature with all the target classes. The frequency table contains a count of each class name associated with every distinct feature value.

b) Find the Rule for each feature with the help of a frequency table. Choose among the classes to generate a Rule (Feature Value → Class Value) depending on the maximum value of that frequency.

c) Following Find the error rate of each Rule for the class value of that feature that does not agree with that Rule.

d) Generate a rank list by ordering the value of symmetrical uncertainty of each feature in descending order.

4) *RELIEF*: Marko Robnik-ikonja and Igor Kononenko describe the RELIEF method, which uses instance-based learning to assign each feature a pertinent weight [20]. The Relief algorithm's main principle is to treat each feature as an independent entity and determine the importance of the feature based on its potential to discriminate data samples that are close to one another. The Relief algorithm has the inherent flaw of giving high relevance ratings to all the discriminating features, although some features have substantial correlations. The algorithm consists of three main parts, which followed as:

a) Determine the distance between the closest hit and miss.
b) Determine each feature's weight and update it following dataset samples.
c) Retrieve the top k features or an ordered list of features based on a specified threshold.

5) *Gain Ratio:* The information Gain metric favors tests having a wide range of outcomes. The Information Gain prefers attributes with more possible values, even though features with fewer values are more informative. The gain ratio (GR), an addition to Information Gain, is used by C4.5, a method that replaces the ID3 algorithm to combat bias [21]. By using the intrinsic information (Entropy) of that characteristic, it corrects Information Gain. Let D be a collection of d data samples divided into different classes. The anticipated details required to categorize a particular data sample are provided by,

$$\text{GainRatio}\left(\text{feature}_i\right) = \frac{\text{Gain}\left(\text{feature}_i\right)}{\text{splitInfo}_{\text{feature}_j}(D)} \tag{6}$$

The steps to find the Gain Ratio are as follows:

a) Find Information Gain of each feature.
b) Find Discover the split information for each feature. This value shows the possible data generated by dividing the data set used for training, D, into v divisions, which coincide with the v results of a test on feature i.

$$\text{SplitInfo}_{\text{feature}_i}(\text{D}) = -\sum_{j=1}^{v} \frac{|\text{D}_j|}{|\text{D}|} \times \log_2\left(\frac{|\text{D}_j|}{|\text{D}|}\right) \tag{7}$$

c) Find the Gain Ratio of each feature according to its equation.
d) Generate a rank list by ordering the value of the Gain Ratio of each feature in descending order.

6) *Chi-Square:* The chi-square test of independence allows the researcher to determine whether features are independent of one another or whether there is a pattern of dependence between them [22]. The researcher might assert there exists a statistically significant connection between the two features when there is a dependency. The chi-square calculation in the feature selection scenario is as follows:

$$\chi^2_{(x_i)} = \sum_{ij} \frac{\left(\text{observed}_{ij} - \text{Expected}_{ij}\right)^2}{\text{Expected}_{ij}} \tag{8}$$

where the expected number of features X_i computed as:

$$E_{ij} = \frac{total\_count(Class_j) * total\_count(Feature_i)}{total\_samples} \tag{9}$$

The chi-squared test creates a frequency table containing the class value for each feature and each unique feature value. The frequency table determines the anticipated value for each distinct value of that particular feature. We need to count the chi-squared value for each feature with class to select features. A feature with a high chi-squared value for the target class indicates a strong relationship with the class. Arrange all chi-squared values in descending order to generate a rank list.

B. *Rank Aggregation Techniques*

1) *Borda Count (BC) :* A feature's average position in the rank list is its Borda count.

$$Borda\ (i) = \sum_{j=1}^{n} \pi_j(f_i) \tag{10}$$

where The rank of feature f_i in ranking P_{ij} is expressed as $P_{ij}(f_i)$. By arranging Borda counts in ascending order, the Borda algorithm provides each feature a rank [23].

2) *Condorcet :* A The Condorcet technique incorporates a pairwise comparison of the rankings of two features. The Condorcet criterion states that a candidate is the "Condorcet winner" if it outperforms all others in straightforward pairwise comparisons [16]. Depending on this characteristic, the Condorcet aggregation method operates as follows: for each input ranking method, compare the rank of a feature (fi) with the rank of each other feature, each combination at once, and determine which feature "wins" by having the higher rank. Add up these victories for all ranking techniques while keeping separate totals for each pairwise combination. The feature that outperforms all remaining features in pairwise contests is considered the most preferred

feature, and it ranks first in the aggregated ranking. Once we determine the best feature, we can remove the best feature from the feature set and derive another Condorcet winner from the remaining features. We can consider the second Condorcet winner as the second-best feature in the aggregated ranking, and so on. We can use a random tiebreak among winners if we fail to identify a single Condorcet winner.

3) Schulze Method (SSD): Schulze Method is known as Schwartz Sequential Dropping (SSD) [10, 24]. This approach also complies with the Condorcet criterion. We calculate how many ranks of feature fx are higher than feature fy for each pair of features and vice versa. If the first number is higher, then fx defeats fy, else fy defeats fx, and there is a tie if both numbers are equal. We can generate the following graph based on the above information: The features serve as the vertices, and we add an edge from fx to fy whenever fx triumphs over fy. This method guarantees that every created graph contains at least one cycle or a single winning vertex element. The Schwartz set is the collection of all these winning elements. If there are cycles in the Schwartz set, we remove edges from the graph to remove them. If the number of ranks where fx is higher than fy is least, considering all sets of edges in the cycle, we remove the edge that connects fx to fy. We eliminate the elements at once if they receive equal vote counts. The system keeps doing this until there aren't any more cycles in the Schwartz set. The Schwartz set only contains isolated vertices after we broke all of the cycles of the graph, and the associated vertices are the victorious ones. After that, we can delete these vertices from the graph and choose a fresh set of winners. We rate each feature in the aggregation rank based on the order in which they occur as winners. This approach announces only one winner per round. We break ties at random While there are many winners [24].

4) Markov Chain (MC): Dwork et al. explain a method for creating aggregate ranks based on Markov chains(MC4) [16]. The MC4 algorithm is comparable to Google PageRank. The MC4 criteria for feature ranking are as follows: The stationary probability distribution helps the basis for arranging the features to construct the aggregated rating. The Markov chain state pertains to the features to be ranked, and the transition probabilities rely on the input rankings. The algorithm operates as follows: We use the graph to represent the Markov chain. In a graph, every feature correlates to a vertex. A weighted directed edge is generated from the vertex fx to the vertex fy for each pair of vertices fx and fy if fx appears above fy in a ranking and the weight is proportional to the separation between these features in the ranking. The transition probabilities are represented by rescaling these weights to the (0-1) range. If this feature vertex already exists in the graph, then we change the weight by adding it to the original weight, and a number is proportionate to the separation between the two features. We apply the PageRank method to a competing graph production process till convergence. Then, based on each network vertex's "prior importance," we order the attributes in decreasing order.

IV. Proposed Approach

We have proposed an optimized backward feature elimination strategy using heuristic search to extract the most useful subset of features from high dimensional datasets. The suggested method consists of two steps. In step 1, We provide a rank to every feature in the data set using six popular feature ranking techniques. These techniques are Information Gain, Symmetric uncertainly, Gain Ratio, OneR, Relief, and Chi-Square. We then take the rank that each ranking technique provides and combine it using the Schulze rank aggregation method. Schulze method gives a new stable rank to each feature of datasets utilizing the rank provided by all ranking techniques. In phase 2, We arrange all the features of data sets from low-ranked features to high-ranked features. In optimized backward feature elimination, We remove less significant low-ranked features before highly significant high-ranked features. Initially, We consider all the features of datasets and find classification accuracy. Then, we individually remove each feature according to their rank given by the Schulze method from a lower rank to a higher rank. A feature is removed from the final optimized subset of the features if its removal improves the classification accuracy.

To understand the proposed approach, consider the situation shown in Figure 3. Suppose we have a dataset with nine features from f_1 to f_9. The final rank of features based on the Schulze method from low rank to high rank is f_9,f_6, f_7, f_1, f_3, f_5, f_3, f_8, and f_4. Now, someone is trying to eliminate feature f_9. Temp Feature Subset contains features f_1 to f_8. We find Temp Classification Accuracy. If the Temp Classification Accuracy is better than the Best Classification Accuracy, then we will replace the Best Classification Accuracy with Temp Classification Accuracy and remove feature f_9. If the above condition is not satisfied, we will not eliminate feature f_9. We go over each feature of the data set in the same order. In the end, we get the Best Feature Subset. In Figure 3,

feature f_6, f_7, f_3 and f_2 are eliminated and feature f_9, f_1, f_5, f_8 and f_4 are not eliminated by algorithm. This heuristic search procedure considers one feature in each iteration. In the case of n features in data sets, we employ the n times wrapper classification algorithm in the proposed approach.

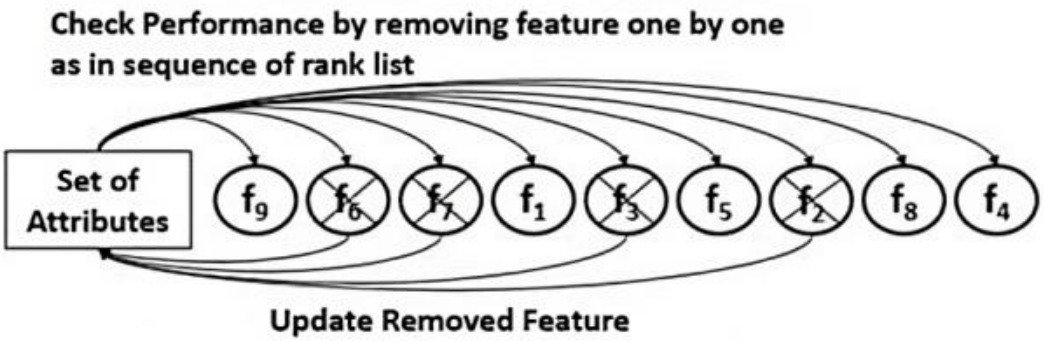


Fig. 3 Optimized Backward Feature Elimination

V. Experiment Results

In this section, we conducted several experiments to examine the effectiveness of the suggested optimized backward feature elimination strategy. We combined the ranks of six independent feature ranking algorithms using the Schulze rank aggregation method. We have used six ranking techniques: Information Gain, Symmetric uncertainly, Gain Ratio, OneR, Relief, and Chi-Squared. We compared the performance of the Schulze method with the above-listed ranking techniques. We have used WEKA (an open-source machine learning library) to find the rank of features for the above listed feature ranking methods with the Naive Bayes classifier as a wrapper classification algorithm [25]. We employed datasets from the UCI data repository shown in Table 1 [26]. Some datasets have a minimal number of data samples and a high number of features. These datasets present challenges to classification and feature selection algorithms. We have used a 10-fold cross-validation classification accuracy measure for the Nave Bayes classifier to compare the performance of various ranking techniques. We have shown a performance comparison of the Schulze method with other ranking techniques using optimized backward feature elimination in Table 2.

Table 1: Accuracy of existing

Dataset	**Number of Features**	**Number of Instances**
Ozon	72	2536
Spambase	57	4601
Waveform v2	40	5000
Coil2000	85	9822
Movement libras	90	360
Semeion	256	1593
Musk	166	476
Isolet	617	1559
Madelon	500	2600
cnae-9	856	1080
multiple-features	649	2000
Micromass	1300	360

VI. Conclusion

Identification of a constrained subset of improved predictive features is critical to the performance of machine learning algorithms. The existence of redundant and irrelevant features in the predictive model leads to poor computational cost and accuracy performance. High-dimensional data sets are not feasible for the most popular search techniques due to their computational requirements. In this research, we created an entirely novel method for selecting features that enables us to employ a hybrid model to discover an optimal set of features for

classification. In this work, we have also introduced the rank aggregation method to generate a stable rank of each attribute across multiple feature ranking methods. Our results showed that rank aggregation methods could improve feature order and feature subset selection. This heuristic search performs exceptionally well on the stable rank list of features compared to the traditional sequential backward elimination search technique. This feature selection technique improves the classification accuracy concerning filter approaches and also the computational cost for the wrapper approach.

Table 2: Experiment Results

	Total Features / Original Accuracy	Removed Features	Improved Accuracy	Total Features / Original Accuracy	Removed Features	Improved Accuracy	Total Features / Original Accuracy	Removed Features	Improved Accuracy
	Ozon			Spambase			Vaveform v2		
IG	72/ 71.76	44	79.02	57/ 79.28	10	80.72	40/ 80	17	82.54
SU		43	79.02		22	89.24		17	82.54
OneR		50	81.11		18	83.54		13	82.17
Relief		41	78.7		16	80.59		14	81.74
Gain Ratio		50	80.91		21	83.09		17	82.54
Chi-Squared		50	81.26		19	90		17	82.54
Schulze Method		45	78.62		20	89.13		17	82.54
Without Rank		33	77.32		24	90.26		15	82.66
	Coil2000			Movement_libras			Semeion		
IG	85/ 78.07	72	92.92	90/ 62.77	22	65	256/ 85.24	36	87.25
SU		75	93.1		23	66.11		38	87.06
OneR		63	91.13		14	65		41	86.62
Relief		66	91.67		21	65.55		48	86.94
Gain Ratio		77	94.03		7	65.55		46	87.82
Chi-Squared		74	92.99		7	64.72		49	86.94
Schulze Method		69	92.98		16	66.11		34	87
Without Rank		77	94.02		19	65.55		38	86.81
	Musk			Isolet			Madelon		
IG	166/ 75.21	87	82.77	617/ 83.77	194	86.08	500/ 59.53	29	61
SU		76	81.3		151	85.76		29	61
OneR		74	81.93		205	86.52		50	61.76
Relief		76	81.72		208	86.33		73	62
Gain Ratio		63	81.09		144	86.01		29	61
Chi-Squared		62	81.3		148	85.95		29	61
Schulze Method		70	82.35		162	85.18		29	61
Without Rank		78	82.33		171	86.65		48	61.42
	cnae-9			multiple-features			Micromass		

IG	856/ 93.14	743	95.27	649/ 95.35	248	96.05	1300/ 94.44	1162	98.05
SU		745	95.27		271	96.15		1162	98.33
OneR		748	94.62		277	96.45		1171	98.05
Relief		743	95.09		220	95.95		1133	98.05
Gain Ratio		740	95.27		283	96.25		1120	97.5
Chi-Squared		743	95.09		240	96.15		1177	97.77
Schulze Method		750	95.27		281	96.1		1177	98.05
Without Rank		739	94.62		193	96.15		1050	97.5

REFERENCES

[1] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, *143*, 106839.

[2] Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

[3] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & electrical engineering, 40(1), 16-28.

[4] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70-79.

[5] Shroff, K. P., & Maheta, H. H. (2015, January). A comparative study of various feature selection techniques in high-dimensional data set to improve classification accuracy. In *2015 international conference on computer communication and informatics (ICCCI)* (pp. 1-6). IEEE

[6] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*(1-2), 273-324.

[7] Shardlow, M. (2016). An analysis of feature selection techniques. *The University of Manchester*, *1*(2016), 1-7.

[8] Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, *1*(1-4), 131-156.

[9] Kalousis, A., Prados, J., & Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems*, *12*, 95-116.

[10] Prati, R. C. (2012, June). Combining feature ranking algorithms through rank aggregation. In *The 2012 international joint conference on neural networks (IJCNN)* (pp. 1-8). Ieee.

[11] Yang, P., Hwa Yang, Y., B Zhou, B., & Y Zomaya, A. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics*, *5*(4), 296-308.

[12] Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, *26*(3), 392-398.

[13] Awada, W., Khoshgoftaar, T. M., Dittman, D., Wald, R., & Napolitano, A. (2012, August). A review of the stability of feature selection techniques for bioinformatics data. In *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)* (pp. 356-363). IEEE.

[14] Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1060-1073.

[15] Wang, H., Khoshgoftaar, T. M., & Gao, K. (2010, August). A comparative study of filter-based feature ranking techniques. In *2010 IEEE international conference on information reuse & integration* (pp. 43-48). IEEE.

[16] Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001, April). Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web* (pp. 613-622).

[17] Li, X., Wang, X., & Xiao, G. (2019). A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications. *Briefings in bioinformatics*, *20*(1), 178-189.

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

[19] Gokalp, O., Tasci, E., & Ugur, A. (2020). A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification. *Expert Systems with Applications*, *146*, 113176.

[20] Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning*, *53*, 23-69.

[21] Prasetiyo, B., Muslim, M. A., & Baroroh, N. (2021, June). Evaluation of feature selection using information gain and gain ratio on bank marketing classification using naïve bayes. In *Journal of physics: conference series* (Vol. 1918, No. 4, p. 042153). IOP Publishing.

[22] Bahassine, S., Madani, A., Al-Sarem, M., & Kissi, M. (2020). Feature selection using an improved Chi-square for Arabic text classification. *Journal of King Saud University-Computer and Information Sciences*, *32*(2), 225-231.

[23] Tang, Y., & Tong, Q. (2016, June). BordaRank: A ranking aggregation based approach to collaborative filtering. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (pp. 1-6). IEEE.

[24] Csar, T., Lackner, M., & Pichler, R. (2018, July). Computing the Schulze Method for Large-Scale Preference Data Sets. In *IJCAI* (pp. 180-187).

[25] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10-18.

[26] Dua, D., & Graff, C. (2017). UCI machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, *7*(1), 62.