Vedatrayee Chatterjee[1]

Dr. Kamal Dhanda[2]

Optimization for Best Feature Selection on Microarray Gene Expression Data

Abstract: - Feature selection serves as a crucial technique in data analysis, eliminating unnecessary elements from a dataset to enhance computational efficiency and improve the accuracy of machine learning models. This study introduces a novel method called Rotate Left and Complement (RLC) for feature selection, employing T statistics to identify informative genes. The RLC algorithm, based on the top m informative genes, presents a promising solution to refine feature sets. The accuracy of categorization is evaluated using the KNN method across three diverse datasets, demonstrating the effectiveness of the proposed approach in optimizing feature selection and contributing to the advancement of machine learning methodologies.

Keywords: Feature selection, Rotate Left and Complement (RLC), KNN, classification

1. Introduction

A distinctive and measurable characteristic of an entity is referred to as a feature, setting it apart from related entities. Feature selection involves the process of choosing a set of attributes that is most relevant or beneficial for a specific problem, either through automatic or manual methods. This practice is also known as variable selection or attribute selection.

Recent technological progress has led to the widespread availability of extensive and highly dimensional datasets on the internet. Despite advancements in computational technology, deriving meaningful insights from these datasets remains a challenging task. The size of a dataset is often characterized by its number of features (N) and instances (P), both of which can be exceptionally large [1]. Particularly in the medical and healthcare domains, professionals face difficulties in rapidly interpreting such vast amounts of data to deliver timely diagnoses, prognoses, and treatment plans. Consequently, data mining has become indispensable in the fields of medical and healthcare.

The presence of unnecessary attributes can lead modeling algorithms astray. Instance-based techniques such as k-nearest neighbor [2] rely on small neighborhoods in the attribute space to make classification and regression predictions. The inclusion of redundant attributes can significantly distort these predictions. Retaining superfluous attributes in the dataset may result in overfitting, where the model learns the training data too well, compromising its predictive accuracy and overall strength. Four categories of feature subsets have been identified:

(a) Completely irrelevant and noisy features

(b) Weakly relevant and redundant features

(c) Weakly relevant and non-redundant features

(d) Strongly relevant features[20]

Computational molecular biology is an interdisciplinary field that integrates computer science, biology, and information technology [4]. The primary goal of this field is to enhance scientific discovery and develop innovative analytical tools for molecular biology. The emergence of DNA microarray datasets has sparked a new area of research in bioinformatics [5]. Microarrays, a relatively recent technology, are employed for treating

[1] 1*Research Scholar, Department of Computer Science and Engineering, School of Engineering & Technology, Om Sterling Global University, Hisar 125001, INDIA.

[2] Associate Professor, Department of Computer Science and Engineering, School of Engineering & Technology, Om Sterling Global University, Hisar 125001, INDIA.

conditions like mouth lesions and conducting pharmacological studies on cancer. These devices consist of a microscope slide printed with thousands of tiny dots in precise locations, typically made from glass, silicon chips, or a nylon membrane. One specific type is the DNA microarray, also known as a gene chip, DNA chip, or biochip, which either monitors DNA or incorporates DNA into its detection mechanism. Each location on the array contains an organized gene or a recognized DNA sequence.

To identify relevant genes, samples from both normal and malignant tissues of patients are collected, and the position and order of each location are recorded in a database. This database becomes a valuable resource for diagnostic tools, aiding in distinguishing between cancer types or identifying healthy and malignant tissues [6]. Typically, these databases contain fewer than 100 samples but encompass a vast number of features, ranging from 6000 to 60000 [5]. Numerous studies have indicated that a considerable portion of genes in DNA microarray experiments may not be pertinent for accurately classifying distinct groups of the problem [7]. Moreover, the utilization of this data with a classifier may lead to overfitting due to the presence of fewer samples compared to the number of genes in the dataset [8].

To address this challenge, a precise feature selection technique is employed to simplify the feature space and identify a set of highly distinctive genes before the classification process [9][10]. Research has demonstrated that a significant portion of genes assessed in DNA microarray experiments doesn't contribute significantly to enhancing classification accuracy across various classes [11].

Hence, the inclusion of feature (gene) selection is crucial for the classification procedure to precisely analyze gene expression profiles [12]. In the context of gene expression data, the application of feature selection is sometimes termed gene selection. Gene selection becomes particularly vital for diagnosing primary tumors and cancer, ultimately contributing to improved medical care.

Feature selection serves various purposes, including minimizing measurement costs, enhancing classifier accuracy, reducing complexity, lowering associated computational expenses, and improving accuracy by eliminating redundant and unnecessary information. A crucial step in data preparation involves feature selection to decrease the dataset's dimensionality [3]. Therefore, implementing feature selection on a dataset offers the following key benefits:

(a) It speeds up algorithm training.

(b) It simplifies and facilitates the interpretation of a model by reducing its complexity.

(c) It improves the accuracy of a model if the right subset is chosen.

(d) It lessens overfitting since there are less duplicate data points, which decrease the chance of making noise-based conclusions.

Enhancing the accuracy of classification error estimation can be achieved by utilizing a restricted set of features. Consequently, a pivotal aspect in establishing a robust tumor classification system based on gene expression is the reduction of dimensionality in the gene expression data [13].

2. Preliminaries

In this discussion, we have briefly covered the fundamental concepts of diverse feature selection methods, Microarray Gene expression data, the T-Statistics measure, and the KNN classifier.

2.1 Methods of Feature Selection

Numerous methods exist for feature selection, including Information Gain [14], Relief [15], Fisher Score [17], Chi Squares [16], and Lasso [18]. Among the numerous independent variables, or features, defining a data instance (e.g., a patient who may have cancer), tumor markers play a crucial role and are detected in bodily fluids like blood, urine, or stool. These datasets may also include a response variable, or label, indicating the patient's tumor type—whether benign or malignant. In "supervised" feature selection, every data instance in the collection has a known response value. If only certain instances have known response values, it is termed "semi-supervised," posing a unique challenge in feature selection. On the other hand, "unsupervised" feature selection is employed when no data instance has a response value [19].

Supervised feature selection methods are deemed more effective than unsupervised ones as they leverage labeled data [21]. However, many real-world scenarios involve a substantial amount of unlabeled data along with a limited amount of labeled data. In such cases, semi-supervised feature selection techniques come into play, considering both labeled and unlabeled data [22]. Semi-supervised techniques can be further classified into filter, wrapper, and embedding approaches.

2.1.1 Filter Method

Feature selection using filter approaches involves choosing features, irrespective of any machine learning model, based on their statistical characteristics and their correlation with the target variable. The Chi-Squared Test [16] [33] is a useful tool for assessing the independence between features and the target. Mutual Information [34] is a method that selects features with high mutual information scores. In Variance Threshold method [35] features with low variance are removed because they contribute little information. Correlation Coefficient method selects [36] features based on their correlation with the target variable, highlighting those with strong linear or monotonic relationships. The ANOVA F-test [37] finds significant characteristics for numerical data by comparing variances within and between groups. The ReliefF Algorithm [15] [38] assesses feature importance based on its capacity to discern between nearby instances, Information Gain [14] [39] calculates the reduction in entropy from data partitioning based on a given feature, and feature importance scores from tree-based models like Random Forests, which can rank features according to their contribution to the model, are additional helpful techniques. Furthermore, feature selection is accomplished using L1-based feature selection (Lasso), [18] which penalises some coefficients to zero. By concentrating on the most informative features, these strategies can cut computational costs, improve model performance, and reduce the complexity of datasets.

In 1992, Kira and Rendell introduced the Relief algorithm, employing a filter approach for feature selection. In this method, features are ranked through evaluation, and the best ones are selected by assessing each feature independently from the classifier [23]. Filter approaches determine a feature's relevance based on its intrinsic characteristics, leading to the elimination of low-scoring features. The resulting set of features is then provided as input to the classification algorithm. Filter approaches offer several advantages, including speed and simplicity in computation, scalability to high-dimensional datasets, and independence from the specific classification algorithm used. Consequently, feature selection is a one-time process, allowing the assessment of multiple classifiers thereafter [10]. Due to their lack of reliance on any particular learning methodology, filter methods can provide versatile solutions applicable to a variety of classifiers.

2.1.2 Wrapper Method

In order to integrate feature selection with model training, wrapper techniques for feature selection use a predictive model to assess the combination of features and choose the best subset based on model performance. Wrapper methods take into account feature interactions, which can result in higher prediction accuracy than filter approaches. Usually, the procedure entails employing techniques like forward selection, backward elimination, or recursive feature elimination (RFE) [40] to search throughout the space of feature subsets. While backward elimination begins with all characteristics and iteratively removes the least significant ones, forward selection begins with an empty model and adds features one by one that most improve the model's performance. The process of Recursive Feature Elimination (RFE) [41] entails fitting the model and iteratively eliminating the least significant features until the target feature count is attained. Because these techniques involve numerous model training cycles to evaluate each subset of characteristics, they can be computationally demanding, particularly for big datasets. However, wrapper approaches [42] frequently produce better feature subsets appropriate to the chosen machine learning algorithm, improving model resilience and accuracy by taking feature dependencies into account and employing cross-validation to prevent overfitting.

Wrapper Methods generate multiple feature subsets, and each subset is employed to construct a model and train the learning algorithm. The algorithm undergoes testing to identify the optimal subset. Various criteria are employed to select features for these subsets [24]. The most discriminative feature subset is determined by minimizing the prediction error of a specific classifier. This method often yields superior performance outcomes compared to the filter method, as it takes into account feature dependencies and introduces bias directly into the learning algorithm. However, it is less universal than the filter approach, requiring repetition when applying a different learning algorithm [25].

2.1.3 Embedded Method

An "embedded method" occurs when feature selection and classifier design are intricately interconnected [26]. In essence, embedded approaches integrate feature selection with the learning algorithm, assessing not only the relationships between individual input features and the output feature but also locally searching for features that enhance discrimination in specific areas. These methods identify optimal subsets for a given cardinality based on independent criteria [27]. Using the model's performance metrics, embedded feature selection techniques evaluate and choose features at the same time, integrating the feature selection process directly into the model training phase.

By being less computationally demanding than wrappers and taking feature interactions into account more successfully than filters, these approaches combine the benefits of both filters and wrappers. Regularisation techniques like as Lasso [18] (L1 regularisation), Ridge (L2 regularisation), and Elastic Net are examples of embedded methods. These techniques penalise the regression coefficients by pushing some of them to zero, hence performing feature selection during the model fitting process. Decision tree-based algorithms [43] such as Random Forest [44] and Gradient Boosting [45] also provide embedded feature selection by naturally ranking features based on their importance to the model's predictive performance. The process typically involves assessing the contribution of each feature to the model’s prediction accuracy, allowing the model to focus on the most relevant features while disregarding the less informative ones. By integrating feature selection within the model training, embedded methods [46] offer a more streamlined approach that tends to be more efficient and effective, producing models that are not only simpler and faster but also potentially more accurate due to the simultaneous optimization of feature selection and model fitting.

2.2 Microarray Gene Expression Data

In the medical domain, microarrays play a crucial role in generating molecular profiles of patient tissues, offering insights into both healthy and diseased conditions. These profiles contribute to a deeper understanding of various diseases and play a pivotal role in enabling more accurate diagnosis, prognosis, therapy planning, and the discovery of medications [4].

Microarrays [49] are sophisticated laboratory instruments capable of simultaneously identifying hundreds of genes and their expressions. Specifically, DNA microarrays [48] consist of microscope slides that are printed with numerous microscopic dots at predetermined locations, each corresponding to a known gene or DNA sequence. These slides are commonly referred to as DNA chips or gene chips. Gene expression, also known as the transcriptome, involves the collection of messenger RNA (mRNA) transcripts expressed by a specific set of genes. This gene expression is detected by using the DNA molecules affixed to each slide as probes.

A high-dimensional dataset necessary for genomics and biomedical research is produced using microarray gene expression data, [47] which measures thousands of genes' expression levels at once. The process of analysing this data involves multiple important steps: preprocessing, which includes background correction, normalisation, and log transformation to guarantee comparability between arrays; feature selection, which uses t-tests, fold-change analysis, and more sophisticated methods like filter, wrapper, or embedded methods because of the high dimensionality and usually small sample sizes; and biological interpretation, which involves mapping the selected genes to known pathways and databases to comprehend their roles in biological processes or disease mechanisms.

2.3 T-statistics

The T-statistic is a valuable tool for feature selection, particularly in contexts like microarray gene expression data analysis, where the goal is to identify which genes (features) are differentially expressed between two groups. For each feature (e.g., each gene in a microarray dataset), the T-statistic [50] is calculated to compare the means of the feature between two groups (e.g. cancerous or tumor vs. non-cancerous or normal).

Genes that exhibit significantly distinct expressions in tumor and normal tissues can be considered for selection. To determine the extent of gene expression variation between normal and tumor tissues, a straightforward T-statistic measure is employed, as stated in (1) [28]. To be included in the discriminant analysis are the top m genes with the greatest T-statistic.

$$t = \frac{\overline{x1} - \overline{x2}}{\sqrt{\frac{v1}{n1} + \frac{v2}{n2}}} \qquad (1)$$

Here

$\overline{x1}$- Mean of Normal samples

$\overline{x2}$- Mean of Tumor samples

n1 - Normal Sample size

n2 - Tumor Sample size

$v1$ - variance of Normal samples

$v2$ - variance of Tumor samples

2.4 k Nearest Neighbor (kNN) Classifier

The kNN classifier is an instance-based model that functions on the principle that unknown instances can be classified by comparing them to known examples using a distance or similarity metric. In the instance space defined by an appropriate distance function, instances that are farther apart are less likely to belong to the same class compared to instances that are in close proximity.

During the learning phase, the kNN algorithm does not extract information from the training data, deferring generalization until the categorization phase. The classification process involves finding the closest neighbor in the instance space and assigning the unknown instance the same class label as the known neighbor. This approach is commonly referred to as a nearest neighbor classifier [51]. Due to their high local sensitivity, nearest neighbor classifiers are particularly prone to noise in the training set. To create robust models, determining the value of k, where k > 1, and relying on a majority vote for class labeling outcomes becomes essential. If k=1, the object is simply assigned to the class of its closest neighbor. Increasing the value of k results in a less sensitive, smoother function.

Closeness is determined using normal distance measurements, with the distance metric sometimes calculated as one minus the correlation value. For continuous variables, Minkowski, Manhattan, and Euclidean distances are employed, while Hamming distance is applied when dealing with categorical variables. In this context, the distance measure used is the Euclidean distance [29].

2.5 Support Vector Machine (SVM) Classifier

Strong supervised machine learning algorithms like Support Vector Machine (SVM) [52] are usually employed for classification problems. In a high-dimensional space, it operates by determining the best hyperplane to divide data points belonging to various classes. The support vectors are the data points that are closest to the hyperplane, which was selected to maximise the margin between the classes. The kernel trick is a technique that allows SVM [54] to handle both linearly and non-linearly separable data. It works by implicitly mapping the input data into a higher-dimensional space where it can be separable linearly.

In support vector machines (SVM), [53] the optimisation task is to discover the hyperplane that minimises a cost function. This usually requires making a trade-off between maximising margin and minimising classification errors. Given that it is a convex optimisation issue, methods like gradient descent and quadratic programming are frequently used to solve it. Applications for SVM are numerous and span a variety of fields, including banking, bioinformatics, image recognition, and text categorization. Its popularity has grown as a result of its good generalisation to new data and its efficiency in processing high-dimensional data with limited training datasets.

SVM has limitations despite its strength, such as choosing the right hyperparameters for the kernel parameters and the regularisation parameter (C). Large datasets can also provide a scaling challenge, while methods like parallelization and stochastic gradient descent can assist lessen this. In the machine learning arsenal, SVM is still a popular and adaptable algorithm that strikes a compromise between performance, adaptability, and simplicity.

3. Proposed Technique

In this study, we introduce a novel approach for feature selection, presenting an algorithm that is both rapid and suitable for highly distributed and parallel environments.

The algorithm initiates with an initial population of unique solutions. As it progresses, each generation consistently yields a distinct population of solutions, and this iterative process continues until the entire search space is covered. Importantly, no point appears more than once, optimizing execution time. Additionally, the algorithm consistently discovers unique points in the search space, thereby enhancing the likelihood of achieving favorable results.

In our proposed searching technique, the complete search space is divided into several subsets. The number of subsets is determined by the length of the search space. For instance, if the length of the string to be searched is three, resulting in a search space size of 10^3=1000, we divide this space into 170 subsets. Among these, 165 subsets contain six elements each, while the remaining five subsets contain two elements each. The approach to dividing the search space into subsets is explained later. Consequently, the element search is conducted from 170 different search points, allowing the parallel execution of the search process.

The initial search points commence with a series of n zeros. For example, if the search space size is 10^3, the starting search point (referred to as the Generator later) is 000. Subsequent initial search points are derived using a defined algorithm.

The crucial aspects involve determining the starting search points and selecting operators to obtain distinct solutions. This section outlines an illustrative example that provides insights into the investigative work carried out throughout the paper.

The notations used are as below:

$\mathbf{S_i}$ = String of length n

D = Total search Space = $\mathbf{10^n}$

G = Total number of Normal Generators

E = Total number of Exceptional Generators

$\boldsymbol{\alpha_k}$ = Maximum value of normal generator for **k** digit decimal string

A decimal coded string of length **n** is used as a representation of each search point. Each string **S** in the search space **D** is of the form $\mathbf{S}= (\mathbf{d_1}, \mathbf{d_2}\ldots \mathbf{d_n})$, where $\mathbf{d_i \in \{0, 1, 2\ldots 9\}, \forall\ i.}$ The total number of strings in decimal representation is $\mathbf{10^n}$. We propose to find the optimal string(s) among these strings. A finite and distinct sample of initial solutions, each of length **n**, is drawn from **D ($\mathbf{10^n}$)** to form the initial population **P**. To incorporate variation within the solutions, a **R**otate **L**eft and **C**omplement operator (**RLC**) has been used [30] [31] [32]. The **RLC** operator is described below:

Suppose we have a solution $\mathbf{s_i}$ of length is 5 (n = 5) at instance t. Let it be $\mathbf{s_i}$ = 09715. Using the RLC operator, it is possible to generate **2 × n = 2 × 5 = 10** different solutions (including $\boldsymbol{s_i}$) from a single solution $\boldsymbol{s_i}$.

The string **0**9715 produces 9715**9** using RLC operator**.** The underlined portion of the string is shifted 1 position to the left and 9's complement of the left most digit of the old string is placed at the unit position of the new string. Thus the generated strings are 97159, 71590, 15902, 59082, 90284, 02840, 28409, 84097, 40971 over (**2*_n_-1**) iterations. The process of generating strings is stopped when the initial string comes back. Thus from a search string of length *n*, we can obtain (2**n*-1) new distinct search strings using the RLC operator. The string 02341 is called a ***Normal Generator***, as it can generate (2**n*-1) number of distinct strings. If RLC operator is applied on a string generated from the normal generator, then the same strings are generated which have already been generated from the normal generator. For any particular value of ***n***, there are a fixed number of normal generators. A string is said to be an ***Exceptional Generator*** if it does not produce (2*n − 1) different strings by successive application of RLC operator.

We may express the maximum value of the Normal Generator for a string of length n as a function of both the string length (n) and the maximum value of the Normal Generator for a string of length n-1. Assume that the maximum value of the normal generator is max_n, where n is the string's length. The string has a minimum length of 1 and a maximum length of 9. The following formula is used to determine the maximum value normal generator:

- $max_n = 10 * max_{n-1} + 4$ When n>2 and n is an Odd number
- $max_n = 10 * (max_{n-1} + 1) + 4$ When n>2 and n is an Even number

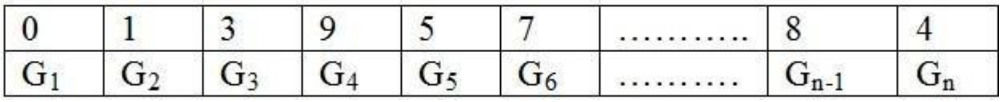
Here is some example:

- $max_0 = 0$
- $max_1 = 10 * max_0 + 4 = 10*0 + 4 = 4$
- $max_2 = 10 * max_1 + 4 = 10*4 + 4 = 44$
- $max_3 = 10 * max_2 + 4 = 10*44 + 4 = 444$
- $max_4 = 10 * (max_3 + 1) + 4 = 10*(444 + 1) + 4 = 4454$ n>3 and n is an even number
- $max_5 = 10 * max_4 + 4 = 10*4454 + 4 = 44544$
- $max_6 = 10 * (max_5 + 1) + 4 = 10*(44544 + 1) + 4 = 445454$ n>3 and n is an even number

3.1 Feature Selection based on RLC

In the initial stage, T-Statistics Measure was applied to three distinct Microarray gene expression cancer datasets to identify genes deemed relevant and highly informative. Subsequently, a set of n top-ranked features was selected to represent chromosomes, with each gene encoded as a decimal number within the range of 0 to 9. Genes with positional values greater than or equal to 5 were retained for classification purposes, while those below 5 were disregarded. The candidate solution, as depicted in Figure 1, underwent Rank-Level Combination (RLC) technique application, resulting in the generation of 2n-1 alternative solutions, contingent upon the string length, i.e., the number of genes represented by n. The initial population was initialized by the first generator. Each population derived through RLC underwent evaluation using a k-Nearest Neighbors (kNN) classifier to ascertain its classification accuracy within the context of cancer gene expression analysis.

Figure 1: Chromosome Representation



0	1	3	9	5	7	………..	8	4
G_1	G_2	G_3	G_4	G_5	G_6	……….	G_{n-1}	G_n

From the above representation G_4, G_5, G_6, ….., G_{n-1} will be selected for classification.

Proposed feature Selection Algorithm:

1. Top scored features are obtained from microarray dataset by applying T-statistics measure.
2. N numbers of informative genes or features are taken as chromosome representation where genes are encoded by decimal numbers.
3. RLC is applied on initial generator to get 2n-1 number of different population.
4. Each population is applied on kNN classifier when gene value is greater than or equal to 5 to get accuracy for each feature subset.
5. Steps 3 to 4 are repeated until maximum value of normal generator is reached or accuracy is reached to 100%.
6. Best feature subset is found when accuracy is most high.

4. Experimental Results

In the initial phase of experimentation, the proposed approach is implemented with the objective of achieving a minimal feature set while maintaining acceptable classification accuracy across three distinct cancer datasets, each comprising samples from two classes. The process commences with the application of T test on the datasets, facilitating the extraction of the most pertinent genes deemed relevant for classification. Diverging from conventional feature selection methodologies, the subsequent step employs Rank-Level Combination (RLC) to further refine the gene pool, effectively reducing the number of features. This strategic reduction aims to enhance classification accuracy by focusing on the most discriminative genes within the dataset. Overall, the approach seeks to strike a balance between feature reduction and classification performance, thereby optimizing the efficiency of cancer classification models.

1. The dataset for colon cancer comprises 62 samples or individuals; of which 22 are normal (non-cancerous) and 40 are tumor tissues (cancerous). There are 2000 genes in the samples. Gene expression numbers are compiled into a 62*2000 matrix, and column indexes are used to identify individual genes.
2. We tested the Prostate Cancer dataset in matrix form (102*12600) further. There are 50 normal (non-cancerous) samples and 52 tumor (cancerous) samples in the collection.
3. We have also tested Leukemia dataset of 72*7129 matrix shape. The 72 samples in the dataset include 25 cases of acute myelogenous leukemia (AML) and 47 cases of acute lymphoblastic leukemia (ALL).

Here 50% of the samples are considered as training data and remaining 50% as test data for all cancer datasets. Table 1 shows three cancer datasets of two- class.

Dataset	**Number of genes or features**	**Samples**	**Class**	
Colon	2000	62	40	Tumor
			22	Normal
Prostate	12600	102	52	Tumor
			50	Normal
Leukemia	7129	72	47	ALL
			25	AML

Table 1: Cancer Datasets

Table 2 shows the informative and relevant genes from T- Statistics measure.

Datasets	**Top Features***
Colon	493, 1423, 249, 377, 765, 245, 267, 66, 14, 822, 1772, 175
Prostate	6185, 10138, 3879, 7520, 4365, 9050, 205, 5654, 3649, 12135, 728, 7768
Leukemia	6854, 758, 1685, 2354, 5171, 5501, 4973, 1909, 4211, 804, 1144, 4680

***** The genes or features are identified by the column number

Table 2: Best features using T Statistics

The best relevant genes are taken as initial generator where each gene has been encoded by decimal number that is 0 to 9. We have applied RLC to get (2*n-1) different populations. Each population has been applied on kNN classifiers where value of *k* is 5.

Table 3 shows the number of genes or features selected by RLC approach and the classification accuracy on three different cancer datasets. The outcome shows that the characteristics our algorithm chose can achieve the desired greater accuracy with the fewest possible features.

Dataset	**Features selected (Index or column number of features)**	**Accuracy (%)**
Colon	1 (1423)	83.87
	2 (245, 822)	83.87
	3 (765, 377, 822)	83.87
Prostate	1 (6185)	85.29
	2 (3879, 10138)	85.21
	3 (6185, 3879, 12153)	91.16
Leukemia	1 (4211)	91.66
	2 (5501, 1909)	94.44
	3 (6854, 1909, 4211)	97.22

Table 3: Selected features and Accuracy

Finally, Table 4 depicts the comparative study of the results obtained by applying genetic algorithm with different classifiers (using 10 features) [29] and the result acquired by applying RLC with kNN classifier (using 3 features).

Dataset	**Methodology**	**Accuracy (%)**
Colon	GA + KNN	75
	GA + SVM	75
	RLC + KNN	**83.87**
Prostate	GA + KNN	78.04
	GA + SVM	78.04
	RLC + KNN	**91.16**
Leukemia	GA + KNN	72.72
	GA + SVM	72.72
	RLC + KNN	**97.22**

Table 4: Comparison of RLC+KNN classification accuracy with GA

5. Conclusion and Future Scope

In our research, we integrated the Rank-Level Combination (RLC) approach with the k-Nearest Neighbors (KNN) classifier to analyze relevant genes identified through T statistics across three diverse datasets. Our investigation encompassed datasets representing Colon cancer, Prostate cancer, and Leukemia. Notably, our findings demonstrate compelling results wherein the utilization of just one feature yielded notable accuracies. Specifically, for Colon cancer, employing column number 1423 resulted in an accuracy of 83.87%. Similarly, in the context of Prostate cancer, a single feature (column number 6185) achieved an accuracy of 85.29%, while for Leukemia, column number 4211 exhibited remarkable accuracy, reaching 91.66%. These outcomes underscore the efficacy of our approach in pinpointing highly discriminatory genes crucial for accurate classification across different cancer types, thus offering valuable insights for further research and clinical applications.

The findings, succinctly presented in Table 4, underscore the superior performance achieved through the Rank-Level Combination (RLC) approach coupled with the k-Nearest Neighbors (KNN) classifier, particularly in the context of feature reduction, when contrasted with the genetic algorithm in tandem with both KNN and Support Vector Machine (SVM) classifiers. Notably, our methodology showcases promising potential for extension to various other feature selection models, including but not limited to F-Test, Information Gain, and Signal-to-Noise Ratio (SNR). Furthermore, our approach lends itself well to further exploration through integration with SVM classifiers, offering avenues for enhanced classification accuracy and robustness. These insights highlight the versatility and efficacy of our approach in facilitating comprehensive and efficient cancer classification, thus paving the way for continued advancements in this critical domain of research and clinical practice.

References

[1] Liu, H., & Motoda, H. (1998). Feature Selection for Knowledge Discovery and Data Mining, doi:10.1007/978-1-4615-5689-3

[2] Novakovic, Jasmina & Veljovic, Alempije & Ilic, Sinisa & Papic, Milos. (2016). EXPERIMENTAL STUDY OF USING THE K-NEAREST NEIGHBOUR CLASSIFIER WITH FILTER METHODS.

[3] A Review Paper on Feature Selection Methodologies and Their Applications, Shweta Srivastava, Nikita Joshi, Madhvi Gaur, International Journal of Engineering Research and Development e-ISSN: 2278- 067X, p-ISSN: 2278-800X, www.ijerd.com Volume 7, Issue 6 (June 2013), PP. 57-61

[4] M. Banerjee, S. Mitra, and H. Banka, “Evolutionary rough feature selection in gene expression data,” IEEE Trans. Syst., Man, Cybern. C, Appl.Rev., vol. 37, no. 4, pp. 622–632, Jul. 2007.

[5] Bolo´n-Canedo, Vero´nica & Sa´nchez-Maron˜o, Noelia & Alonso-Betanzos, Amparo & Ben´ıtez, Jose´ & Herrera, Francisco. (2014). A review of microarray datasets and applied feature selection methods. In- formation Sciences. 282. 111–135. 10.1016/j.ins.2014.05.042.

[6] S. H. Bouazza, N. Hamdi, A. Zeroual and K. Auhmani, ”Gene-expression-based cancer classification through feature selection with KNN and SVM classifiers,” 2015 Intelligent Systems and Computer Vision (ISCV), Fez, 2015, pp. 1-6. doi: 10.1109/ISACV.2015.7106168

[7] Seijo-Pardo, Borja & Bolo´n-Canedo, Vero´nica & Alonso-Betanzos, Amparo. (2016). Using a feature selection ensemble on DNA microarray datasets. Proceeding of $24^{T\ h}$ European Symposium on Artificial Neural Networks. 277-282.

[8] Kumar, Ammu & Valsala, Preeja. (2013). Feature Selection for high Dimensional DNA Microarray data using hybrid approaches. Bioinformation. 9. 824-8. 10.6026/97320630009824.

[9] Piatetsky-Shapiro, G.; Tamayo, P. Microarray Data Mining: Facing the Challenges. SIGKDD Explor. Newsl. 2003, 5, 1–5.

[10] Saeys, Y.; Inza, I.; Larran˜aga, P. A review of feature selection techniques in bioinformatics. Bioinformat- ics 2007, 23, 2507–2517.

[11] Golub, T.R.; Slonim, D.K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J.P.; Coller, H.; Loh, M.L.; Downing, J.R.; Caligiuri, M.A.; et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring.Science 1999, 286, 531–537.

[12] N. Almugren and H. Alshamlan, “A Survey on Hybrid Feature Selection Methods in Microarray Gene Expression Data for Cancer Classification” in IEEE Access, vol. 7, pp. 78533-78548, 2019. doi: 10.1109/ACCESS.2019.2922987

[13] Liu, S., Xu, C., Zhang, Y. et al. Feature selection of gene expression data for Cancer classification using double RBF-kernels. BMC Bioinformatics 19, 396 (2018) doi:10.1186/s12859-018-2400-2

[14] S. Lei, ”A Feature Selection Method Based on Information Gain and Genetic Algorithm,” 2012 Inter- national Conference on Computer Science and Electronics Engineering, Hangzhou, 2012, pp. 355-358. doi: 10.1109/ICCSEE.2012.97

[15] Kira, Kenji and Rendell, Larry (1992) A Practical Approach to Feature Selection, Proceedings of the Ninth International Workshop on Machine Learning, p249-256.

[16] Jin, Xin & Xu, Anbang & Bie, Rongfang & Guo, Ping. (2006). Machine Learning Techniques and Chi- Square Feature Selection for Cancer Classification Using SAGE Gene Expression Profiles. Lect Notes Comput Sci. 3916. 106-115. 10.1007/11691730 11.

[17] Gu, Quanquan & Li, Zhenhui & Han, Jiawei. (2012). Generalized Fisher Score for Feature Selection. Proceedings of

the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011.

[18] Fonti V, Belitser E. Feature selection using lasso. VU Amsterdam Research Paper in Business Analytics. 2017 Mar 30.

[19] Huang SH. Supervised feature selection: A tutorial. Artif. Intell. Research. 2015 Apr; 4(2):22-37.

[20] Yu, Lei, and Huan Liu. ”Efficient feature selection via analysis of relevance and redundancy.” Journal of machine learning research 5.Oct (2004): 1205-1224.

[21] Sheikhpour, Razieh, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. ”A survey on semi-supervised feature selection methods.” Pattern Recognition 64 (2017): 141-158.

[22] Zhao, Zheng, and Huan Liu. “Semi-supervised feature selection via spectral analysis.” In Proceedings of the 2007 SIAM international conference on data mining, pp. 641-646. Society for Industrial and Applied Mathematics, 2007.

[23] Karabulut, Esra Mahsereci, Selma Ays¸e O¨ zel, and Turgay Ibrikci.“ A comparative study on the effect of feature selection on classification accuracy.” Procedia Technology 1 (2012): 323-327.

[24] Fonti, Valeria, and Eduard Belitser. “Feature selection using lasso.” VU Amsterdam Research Paper in Business Analytics (2017): 1-25.

[25] Ang, Jun Chin, et al. “Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection.” IEEE/ACM transactions on computational biology and bioinformatics 13.5 (2015): 971-989.

[26] Hua, Jianping, Waibhav D. Tembe, and Edward R. Dougherty. “Performance of feature-selection meth- ods in the classification of high-dimension data.” Pattern Recognition 42.3 (2009): 409-424.

[27] Kumar, Vipin, and Sonajharia Minz. “Feature selection: a literature review.” SmartCR 4.3 (2014): 211- 229.

[28] K. Yendrapalli, R. Basnet, S. Mukkamala, and A.H. Sung, “Gene Selection for Tumor Classification Using Microarray Gene Expression Data,” in Proceedings of the World Congress on Engineering, vol. I, 2007.

[29] Gunavathi, C., and K. Premalatha. “Performance analysis of genetic algorithm with kNN and SVM for feature selection in tumor classification.” Int J Comput Electr Autom Control Inf Eng 8.8 (2014): 1490-7.

[30] B. Brey, Intel microprocessors: 8086/8088, 80186, 80286, 80386, and 80486 architecture, programming and interfacing. New Delhi: Prentice Hall, 1995.

[31] Manaar Alam, Soumyajit Chatterjee, Haider Banka A Novel Parallel Search Technique for Optimization 2016 3rd International Conference on Recent Advances in Information Technology (RAIT).

[32] V Chatterjee, Dr. Kamal Dhanda, “A Novel Parallel Search Technique To Optimize Benchmark Functions For 10n Search Points”, Corrosion and Protection, Journal of Material Engineering VOL 51 (Issue 1), pp 529-542.

[33] Zhai, Yujia & Song, Wei & Liu, Xianjun & Liu, Lizhen & Zhao, Xinlei. (2018). A Chi-Square Statistics Based Feature Selection Method in Text Classification. 160-163. 10.1109/ICSESS.2018.8663882.

[34] M. A. Sulaiman and J. Labadin, "Feature selection based on mutual information," 2015 9th International Conference on IT in Asia (CITA), Sarawak, Malaysia, 2015, pp. 1-6, doi: 10.1109/CITA.2015.7349827.

[35] M. Al Fatih Abil Fida, T. Ahmad and M. Ntahobari, "Variance Threshold as Early Screening to Boruta Feature Selection for Intrusion Detection System," 2021 13th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2021, pp. 46-50, doi: 10.1109/ICTS52701.2021.9608852.

[36] Zhou, Hongfang & Wang, Xiqian & Zhu, Rourou. (2022). Feature selection based on mutual information with correlation coefficient. Applied Intelligence. 52. 1-18. 10.1007/s10489-021-02524-x.

[37] Elssied, Nadir & Ibrahim, Assoc Prof. Dr. Othman & Hamza Osman, Ahmed. (2014). A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification. Research Journal of Applied Sciences, Engineering and Technology. 7. 625-638. 10.19026/rjaset.7.299.

[38] Urbanowicz RJ, Olson RS, Schmitt P, Meeker M, Moore JH. Benchmarking relief-based feature selection methods for bioinformatics data mining. J Biomed Inform. 2018 Sep;85:168-188. doi: 10.1016/j.jbi.2018.07.015. Epub 2018 Jul 17. PMID: 30030120; PMCID: PMC6299838.

[39] P. V. Agrawal and D. D. Kshirsagar, "Information Gain-based Feature Selection Method in Malware Detection for MalDroid2020," 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), Villupuram, India, 2022, pp. 1-5, doi: 10.1109/ICSTSN53084.2022.9761336.

[40] Awad M, Fraihat S. Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems. Journal of Sensor and Actuator Networks. 2023; 12(5):67. https://doi.org/10.3390/jsan12050067.

[41] X. Zeng, Y. -W. Chen and C. Tao, "Feature Selection Using Recursive Feature Elimination for Handwritten Digit Recognition," 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 2009, pp. 1205-1208, doi: 10.1109/IIH-MSP.2009.145.

[42] Mao Y, Yang Y. A Wrapper Feature Subset Selection Method Based on Randomized Search and Multilayer Structure. Biomed Res Int. 2019 Nov 4;2019:9864213. doi: 10.1155/2019/9864213. PMID: 31828154; PMCID: PMC6885241.

[43] H. Liu, M. Cocea and W. Ding, "Decision tree learning based feature evaluation and selection for image classification," 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 2017,

pp. 569-574, doi: 10.1109/ICMLC.2017.8108975.

[44] M. T. Uddin and M. A. Uddiny, "A guided random forest based feature selection approach for activity recognition," 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Savar, Bangladesh, 2015, pp. 1-6, doi: 10.1109/ICEEICT.2015.7307376.

[45] V. Chatterjee, A. Maitra, S. Ghosh, H. Banerjee, S. Puitandi, and A. Mukherjee, “An efficient approach for breast cancer classification using machine learning”, J. Decis. Anal. Int. Comp., vol. 4, no. 1, pp. 32–46, Jan. 2024.

[46] Naik AK, Kuppili V. An embedded feature selection method based on generalized classifier neural network for cancer classification. Comput Biol Med. 2024 Jan;168:107677. doi: 10.1016/j.compbiomed.2023.107677. Epub 2023 Nov 8. PMID: 37988786.

[47] Ahmad Zamri N, Ab. Aziz NA, Bhuvaneswari T, Abdul Aziz NH, Ghazali AK. Feature Selection of Microarray Data Using Simulated Kalman Filter with Mutation. Processes. 2023; 11(8):2409. https://doi.org/10.3390/pr11082409.

[48] A. Anaissi, P. J. Kennedy and M. Goyal, "Feature Selection of Imbalanced Gene Expression Microarray Data," 2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Sydney, NSW, Australia, 2011, pp. 73-78, doi: 10.1109/SNPD.2011.12.

[49] Alhenawi E, Al-Sayyed R, Hudaib A, Mirjalili S. Feature selection methods on gene expression microarray data for cancer classification: A systematic review. Comput Biol Med. 2022 Jan;140:105051. doi: 10.1016/j.compbiomed.2021.105051. Epub 2021 Nov 23. PMID: 34839186.

[50] For each feature (e.g., each gene in a microarray dataset), calculate the t-statistic to compare the means of the feature between two groups (e.g., diseased vs. healthy).

[51] S. Begum, D. Chakraborty and R. Sarkar, "Data Classification Using Feature Selection and kNN Machine Learning Approach," 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 2015, pp. 811-814, doi: 10.1109/CICN.2015.165.

[52] O. Strub, "Optimal Feature Selection for Support Vector Machine Classifiers," 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, Singapore, 2020, pp. 304-308, doi: 10.1109/IEEM45057.2020.9309859.

[53] Bhat, Ashwini & Krishna, Vijaya. (2020). Feature Selection For Indian Instrument Recognition Using SVM Classifier. 277-280. 10.1109/ICIEM48762.2020.9160223.

[54] Ravi Kumar, G. & Ramachandra, G & Nagamani, K. (2014). An Efficient Feature Selection System to Integrating SVM with Genetic Algorithm for Large Medical Datasets. International Journal of Advanced Research in Computer Science and Software Engineering. 4. 272-277.