Academic Journal of Nawroz University (AJNU), Vol.11, No.4, 2022

https://doi.org/10.25007/ajnu.v11n4a1424

The Effect of Data Splitting Methods on Classification Performance in Wrapper-Based Cuttlefish Gene-Selection Model

Wahab Kh. Arabo[1], Omar M. Malallah[2]

[1,2] Department of Computer Science, University of Zakho, Duhok, Iraq

ABSTRACT

Considering the high dimensionality of gene expression datasets, selecting informative genes is key to improving classification performance. The outcomes of data classification, on the other hand, are affected by data splitting strategies for the training-testing task. In light of the above facts, this paper aims to investigate the impact of three different data splitting methods on the performance of eight well-known classifiers when paired by Cuttlefish algorithm (CFA) as a Gene-Selection. The classification algorithms included in this study are K-Nearest Neighbors (KNN), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Linear Support Vector Machine (SVM-L), Sigmoid Support Vector Machine (SVM-S), Random Forest (RF), Decision Tree (DT), and Linear Discriminant Analysis (LDA). Whereas the tested data splitting methods are cross-validation (CV), train-test (TT), and train-validation-test (TVT). The efficacy of the investigated classifiers was evaluated on nine cancer gene expression datasets using various evaluation metrics, such as accuracy, F1-score, Friedman test. Experimental results revealed that LDA and SVM-L outperformed other algorithms in general. In contrast, the RF and DT algorithms provided the worst results. In most often used datasets, the results of all algorithms demonstrated that the train-test method of data separation is more accurate than the train-validation-test method, while the cross-validation method was superior to both. Furthermore, RF and GNB was affected by data splitting techniques less than other classifiers, whereas the LDA was the most affected one.

KEY WORDS: Cuttlefish Optimization Algorithm, Gene-Selection, Gene Expression, Classification, Cross-Validation.

1. Introduction

The biotechnology of deoxyribonucleic acid (DNA) microarray has advanced significantly in recent years, allowing scientists to screen a large number of genes at the same time for diverse reasons. Oncologists can use gene expression data analysis to aid in the classification of various types of cancer, thus facilitating treatment choices. The nature of this type of data, however, poses some difficult challenges that researchers must consider to address prior to the analysis/ prediction process. The key challenge of this data is their high dimensionality. More specifically, the available microarray datasets typically contain much smaller number of samples when compared to their number of attributes. As a result, this condition makes the learning process slow and might lead to misclassification (Fahrudin, Syarif, and Barakbah 2017) (Shi et al. 2021)(Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2013). One of the most common approaches to addressing the above issues is gene selection (Vafaee, Mosafer, and Hossein 2016a).

Gene selection is a branch of feature selection in which subsets of relevant and significant genes are selected with the aim of improving the performance of a classification or prediction model. According to the literature, methods based on gene selection are classified into four categories: Filter, Wrapper, Embedding, and Hybrid. In filter methods, the significance of each gene or set of genes is scored using some statistical measures, such as ANOVA, Chi-Squared, and so on. These methods are simple to apply to high-dimensional datasets, have a low degree of complexity, and are classifier-independent. On the other hand, in wrapper methods, a subset of genes is selected using some stochastic and meta-heuristic optimization methods, such as Cuttlefish Algorithm (CFA), Random Forest (RF), Decision Tree (DT), and so on, in which those selected genes are evaluated using a particular classifier algorithm. Although the performance of these techniques is outstanding, the search space complexity is relatively high for situations with millions of genes, resulting in increased time complexity. Unlike previous methods, embedded methods select the most significant genes based on the classifier properties. The last category, hybrid methods, seeks to integrate more than one category of gene selection by concentrating on their mutual benefits (Fahrudin, Syarif, and Barakbah 2017)(Vafaee, Mosafer, and Hossein 2016b).

In this paper, the impact of three data splitting strategies on the classification performance of gene expression data are studied. More specifically, a

number of wrapper-based Cuttlefish models are used for meeting the above purpose. Here, a subset of genes is selected by CFA, and then those genes are evaluated independently by several classifiers. The classifiers investigated are: K-Nearest Neighbors (KNN), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Linear Support Vector Machine (SVM-L), Sigmoid Support Vector Machine (SVM-S), RF, DT, and Linear Discriminant Analysis (LDA).

The rest of this paper is organized as follows: Section 2 includes the related works. Section 3 presents the methodology of this study. Section 4 and 5 present and discus the experimental evaluation and results. While Section 6 contains the conclusions.

2. RELATED WORKS

Recently, cancer gene expression datasets have attracted the attention of many researches, where numerous of wrapper-based approaches were utilized to address the problem of selecting the most useful genes. For example, in (Zhu, Ong, and Dash 2007) Zhu et al. presented a new gene selection approach that makes use of filter and wrapper methods. They used a cross-entropy based technique called (Markov blanket) for removing redundant and irrelevant genes. Then, they used Genetic Algorithm (GA) to further select salient genes from the filtered genes so that better accuracy rate can achieved compared to the recent existing approaches. Lee et al. (Lee and Leu 2011) used GA with dynamic parameter setting to provide several subsets of genes, then they ranked the genes based on their occurrence frequencies. While, they compared the efficiency of the selected genes using SVM. They claimed that their method performs better in terms of the number of selected genes and the prediction accuracy when compared with the existing state-of-the-art approaches.

In (Soufan et al. 2015), Soufan et al. developed a web-based tool that efficiently select best features for a variety of problems including gene selection. Their proposed tool is based on a wrapper paradigm, which uses parallel GA to examine and evaluate the candidate collections of features. Furthermore, in (Begum et al. 2018), Begum et al. showed that Memetic algorithm (MA) outperforms GA, simulated annealing (SA), and tabu search (TS) in selecting genes from three cancer microarray datasets. While in (Sayed et al. 2019), Sayed et al. presented a new feature selection method based on t-test and GA, Where the data was preprocessed using t-test, then a nested GA was used to obtain the most valuable subset of features by assembling data from two different datasets. In (Jansi Rani and Devaraj 2019), Jansi et al. used mutual information and GA as two-phase hybrid gene selection approach for classifying two cancer datasets. The most valuable genes were selected in the first phase were passed to GA in the second phase. Similar approach was established in (Pragadeesh et al. 2019) by Pragadeesh et al., where Information Gain (IG) was used to remove redundant genes that will not contribute in the final classification, following that, GA was employed to find the best minimal subset of required genes. Both (Jansi Rani and Devaraj 2019) and (Pragadeesh et al. 2019) utilized the SVM classifier to evaluate the efficiency of their selected genes subsets.

The Particle Swarm Optimization (PSO) was used by a number of researchers to provide solutions to gene selection challenges. For example, Alba (Alba et al. 2007) compared the utilization of PSO and genetic algorithm (GA) as a gene selection for high-dimensional microarray data, both evaluated with SVM classifier. On six publicly available cancer datasets, a modified PSO, called Geometric PSO, was presented for comparison with the GA. In another study, Mohamad et al. (Mohamad et al. 2009) proposed a new PSO named Improved binary PSO combined with SVM classifier to select a near-optimal subset of informative genes relevant to cancer classification. The existing rule for updating the particle position and velocity was modified.

In Ref (Chen et al. 2014), Chen et al. proposed a novel method for gene selection using PSO with a decision tree as the classifier to select a minimum number of relevant genes from the genes in the dataset that can help identify cancers. In (Sahu and Mishra 2012), SVM, KNN, and Probabilistic Neural Network (PNN) were used by Sahu et al. to evaluate the subset of selected genes. This work was performed in two steps, start with using signal-to-noise ratio (SNR) filtering technique, followed by selecting optimal subset of genes using PSO. Furthermore, PSO with adaptive KNN gene selection technique was proposed by Kar et al. (Kar, Sharma, and Maitra 2015a) to select a small subset of relevant genes that are adequate for the classification goal.

A novel gene selection method named Gene Selection Programming (GSP) was proposed by Alanni et al. (Alanni et al. 2019) to select informative genes for better cancer classification. The GSP based model utilizes Gene Expression Programming (GEP) method with a new proposed population initialization algorithm. Moreover, a new fitness function alongside with mutation and recombination operators were modified for better improvement of the model. While SVM with a linear kernel was used as a model classifier.

Wang et al. (Wang et al. 2017) proposed an improved wrapper-based gene selection method by introducing the Markov blanket technique to reduce the required evaluation time by eliminating redundant

genes in the ten well-known publicly available datasets. The selected genes were evaluated utilizing three commonly used classifiers: KNN, Naïve Bayes and C4.5 decision tree. Furthermore, the same classifiers were used to evaluate the goodness of selected genes as well as to evaluate the quality of the final gene subset obtained.

Arshak and Eesa (Arshak and Eesa 2018) proposed a model based on the CFA as a gene selection algorithm to select the most relevant genes. While KNN was used to evaluate the goodness of the resulted genes produced by the CFA. Eight cancer datasets such as Leukemia, Colon, Lung Michigan, Lung Ontario, Breast, Prostate, DLBCL-Harvard, and Central Nervous System were used with the proposed model. Dino et all. (Dino et al. 2022) utilized CFA with Principle Component Analysis to find optimum gene subset in gene expression data classification.

Alshamlan et al. proposed a new model based on Artificial Bee Colony (ABC) to select the informative relevant features for the classification accuracy and SVM for the classification purpose(Alshamlan, Badr, and Alohali 2019). In (Tabakhi et al. 2015) an unsupervised gene selection method was proposed, which incorporates the ant colony optimization algorithm into the filter approach.

Many other methods were proposed in literature for gene selection, such as in (Dash, Dash, and Rautray 2022) a new metaheuristic approach was implemented using binary shuffled frog leaping algorithm with KNN, in (Othman, Kumaran, and Yusuf 2020) a hybrid multi-objective cuckoo search with evolutionary operators, and in (Baliarsingh, Vipsita, and Dash 2020) the enhanced Jaya algorithm and forest optimization algorithm were utilized for the mentioned purpose.

It is worth to mention that the methods of data splitting were various from one research to another, for example, leave one out cross validation were followed in (Alshamlan, Badr, and Alohali 2019; Kar, Sharma, and Maitra 2015b; Sahu and Mishra 2012), while different numbers of cross-validation folds were applied in other studies such as 3 in (Kar, Sharma, and Maitra 2015b) , 4 in (Li, Zhang, and Ogihara 2004), 5 in (Guo et al. 2016; Lee and Leu 2011; Zhu, Ong, and Dash 2007) , also 10 folds are used in some literature such as (Alba et al. 2007; Baliarsingh, Vipsita, and Dash 2020; Begum et al. 2018; Li, Zhang, and Ogihara 2004) . Furthermore, some researchers divide the data into train-validation-test while others just divide it to train-test with different ratio such as (Abdu-Aljabar and Awad 2021; Jansi Rani and Devaraj 2019; Lee and Leu 2011; Ooi and Tan 2003). Since there is variation in the results of all these cases, beside the diversity in the used wrapper-based methods, some valuable questions may be raised up; such as, which classifier get effected more/less by these different data splitting methods? Which classifier is more/less appropriate with gene expression data? Motivated by finding answers to those questions, this study has been established and aims to investigate such issues.

3. METHODOLOGY

3.1 Workflow Steps

In this study, cross-validation (CV), train-test (TT), and train-validation-test (TVT) data splitting strategies were applied to each gene dataset used. For each strategy, the workflow steps are as follows: First, the gene data were standardized using the Z-score method. Second, eight wrapper-based Cuttlefish models were developed for gene selection, namely LR, GNB, SVM-L, SVM-S, RF, DT, and LDA. Finally, the results were compared to each other using the Friedman test to determine which model gets affected mostly/lessely by the data splitting strategies investigated. These steps are illustrated in Fig. 1.

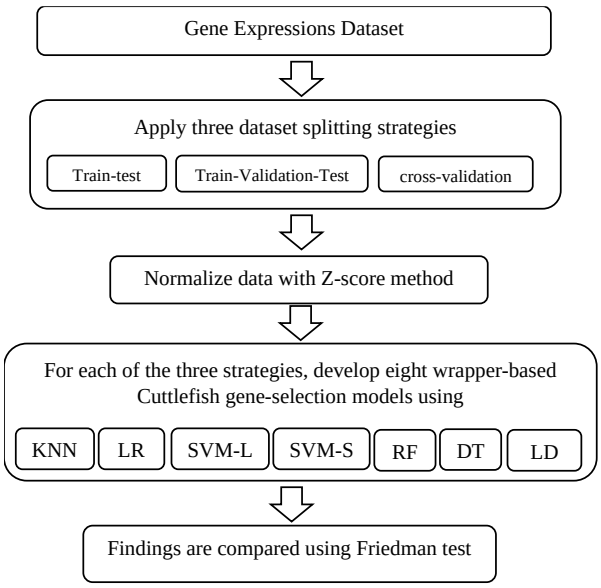


Fig. 1. Diagram of the research methodology

3.2 Gene Selection Based CFA

CFA mimics the process of the cuttlefish's color altering behavior to address the optimization problems. CFA combines two fundamental mechanisms termed *reflection* and *visibility* to imitate this behavior (Sabry Eesa, Mohsin Abdulazeez, and Orman 2013). This combination is formulated into six cases: cases 1,2 and 6 are utilized as global search while the remaining cases are used as local search (Sabry Eesa, Mohsin Abdulazeez, and Orman 2013). The algorithm's developers, in (Eesa, Orman, and Brifcani 2015), employed it as a feature selection with decision tree classifier to find the best feature set for intrusion detection system. The process starts with random initialization, sorting population, finding the highest fitness individual called $AV_{best,}$ then derives a new solution called *best* from $AV_{best,}$ and follows CFA cases

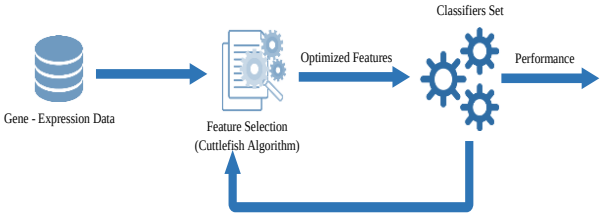
to reach the optimal solution. In this paper the same approach of (Eesa, Orman, and Brifcani 2015) is followed in initialization and ranking individuals' steps, while the CFA cases are slightly modified to better fit the gene expression datasets. Besides, the original one includes the average and best subset with different sizes, while our modified one unifies their sizes. Fig. 2, illustrated the scheme of gene selection approach based on CFA.

Fig. 2. the scheme of gene selection with CFA

The CFA steps used in this work are described as follows:

3.1 Initialization

CFA algorithm starts by preserving the gene locations (indices) of a particular dataset in a $RankedArray$, where $RankedArray = [1,2,\dots,M]$ and M is the gene size. Then it initializes a population (P) with N random solutions. Each solution P_i is linked with *SelectedGenes* and *UnselectedGenes*, where both *SelectedGenes* and *UnselectedGenes* are subsets of



RankedArray, and $SelectedGenes \cap UnselectedGenes = \emptyset$.

To illustrate the algorithm operation, suppose a given dataset consists of 10 genes and the algorithm decides to select 5 of them. The algorithm will create a new array called *RankedArray* = [1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10]. Then, for each solution P_i, it selects 5 genes from the *RankedArray* randomly and assigns them to the *SelectedGenes* subset of P_i, before assigning the rest of the *RankedArray* genes to the *UnselectedGenes* subset of P_i. Assume that the *SelectedGenes* for the first solution (P_0) was chosen randomly to be [5, 3, 9, 7, 2], and the *UnselectedGenes* are [1, 4, 6, 8, 10]. After ranking and initializing, the best solution of the population will be preserved in both AV_{best} and *best*, where AV_{best} and *best* are the two solutions including the best subset of genes *SelectedGenes* and the remaining genes *UnselectedGenes*, respectively. The size of the selected genes (Z) is constant for all solutions in CFA, which is equal to 10% of the whole genes size in the particular dataset. In the following cases of CFA, R is any random number between 0 and Z, and V=Z-R, where R and V represent the parameters of reflection and visibility, respectively.

3.2 Case 1 and 2

In these two cases, the population P is sorted in descending order based on fitness values. Then the first Kth individuals are improved by swapping V of selected genes with unselected. K and R are integer numbers generated randomly between (0, N/2) and (0, 10% of genes total number), respectively. These cases can be formulated as follow:

$Reflection_i = randomSet[R] \subset P_i.SelectedGenes$ (1)
$Visability_i = randomSet[V] \subset P_i.unSelectedGenes$ (2)
$newSubset_i = Reflection_i \cup Visability_i$ (3)

3.3 Case 3 and 4

In these cases, the Best and AV_{best} are improved separately following the same approach of previous cases as follow:

$Reflection_{best} = randomSet[R] \subset Best.SelectedGenes$ (4)
$Visability_{best} = randomSet[V] \subset Best.unSelectedGenes$(5)
$newSubset_{best} = Reflection_{best} \cup Visability_{best}$ (6)
$Reflection_{AVbest} = randomSet[R] \subset AV_{best}.SelectedGenes$ (7)
$Visability_{AVbest} = randomSet[V] \subset AV_{best}.unSelectedGenes$ (8)
$newSubset_{AVbest} = Reflection_{AVbest} \cup Visability_{AVbest}$(9)

It is worth mentioning that the original features selection approach in (Eesa, Orman, and Brifcani 2015) enhanced 25% of the population by replacing some of their features with the *best* selected features. We will use the same procedure in this work.

3.4 Case 5

In this case, the Best and AV_{best} cooperate to find a new solution. The new solution contains R randomly selected *Best* genes and V randomly selected AV_{best} genes. This case can be formulated as follow:

$Reflection_{best} = randomSet[R] \subset Best.SelectedGenes$(10)
$Visability_{AVbest} = randomSet[V] \subset AVbest.SelectedGenes$ (11)
$newSubset_{new} = Reflection_{best} \cup Visability_{AVbest}$ (12)

In (Eesa, Orman, and Brifcani 2015), this case was used to derive *Best* from AV_{best} by reducing the number of selected features. In this work genes of *Best* and AV_{best} are merged partially to produce a new solution.

3.5 Case 6

This case is utilized to produce m solutions using a random generator technique. where $m = N - K$, and K is a previously produced random number from Cases 1 and 2.

In all above cases, if the new generated solution is superior to the present solution, the current solution is dropped in favor of the new one. The population of 20 size is used for 100 epochs. Because of the large numbers of genes in dataset, the local search cases (case 3, 4 and case 5) were repeated 10 times in each epoch.

4. EXPERIMENTAL SETUP

To assess the performance of CFA as gene selection and to evaluate the outcomes goodness for each optimal gene subset provided, eight different machine learning classification models are used. These models are: KNN, LR, GNB, SVM-L, SVM-S, RF, DT, and LDA. On the other hand, to check the efficiency of the classifiers, nine datasets of cancer gene expression are used, which are: Breast, Central Nervous System (CNS), Colon Tumor, Lunge Cancer, Leukemia (2

classes), Leukimia_3c (3 classes), Leukimia_4c (4 classes), MLL and Ovarian Tumor. Table 1 summarizes these datasets information. These datasets can be downloaded from ELVIRA Biomedical Data Set Repository (available at: https://leo.ugr.es/elvira/DBCRepository/index.html).

In addition, the experiments in this research include three different techniques for data splitting: 5 Folds are used in CV, 60% Train and 40% Test are used in TT, and 70% Train, 15% Validation, and 15% Test are used in TVT.

To evaluate the performance of the models constructed from the training data of the gene expression datasets, two performance metrics were used: Accuracy rate and F1-score. Friedman Test was also used to investigate the effect of data splitting method on model performance. This metric (Friedman) was appropriate tool to compare a set of algorithms applied on the same subject, which tells whether there is a significant different between/among the results or not, beside ranking algorithms(Settouti, Bechar, and Chikh 2016).

5. RESULTS AND DISCUSSION

Table 2 demonstrates the models' efficiency in terms of Accuracy and F_1-score for each classifier predictions by using 5-Fold Cross-Validation method to estimate the performance of the used models shown in Fig. 1. Results in Table 2 show that the SVM-L classifier has the best result in the most cases in terms of Accuracy value and F_1-score, followed by LDA as the second-best model, except for one case when using KNN with colon dataset, which gives the best performance.

While Table 3 depicts that the LDA algorithm outperforms all other models, including SVM-L model, in terms of Accuracy and F1-score when using Train-Test method for splitting data, with one exception for LR classifier utilized with Breast cancer dataset, which gives the best results.

TABLE 1.
Summary Information of Cancer Gene Expression Datasets.

Name	No. of instances	Dimensions	No. of classes
Breast	97	24481	2
CNS	60	7129	2
Colon Tumor	62	2000	2
Lung Cancer	203	12600	5
MLL	72	12582	3
Ovarian Cancer	253	15154	2
Leukemia	72	7129	2
Leukemia_3c	72	7129	3
Leukemia_4c	72	7129	4

TABLE 2
5-Folds Cross-Validation Accuracy and F1-score.

	Accuracy									F_1-score								
	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian
KNN	81.32	88.33	**98.46**	97.24	94.38	100	97.55	97.24	96.82	80.78	85.92	**98.3**	96.28	95.3	100	96.56	96.9	96.45
LR	86.68	90	95.13	100	97.14	100	97.52	98.57	100	84.85	**87.05**	94.27	100	97.56	100	96.5	98.52	100
GNB	62.74	83.33	87.31	100	95.9	100	97.55	100	95.68	57.83	77.73	86.79	100	**97.7**	100	94.86	100	95.22
SVM-L	**90.63**	**91.67**	**100**	100	**97.24**	100	**98.04**	98.57	100	**90.38**	83.11	**100**	100	97.66	100	**97.7**	**98.75**	100
SVM-S	78.32	65	88.72	97.14	95.81	100	93.57	100	95.67	77.83	78.74	81.05	96.44	97.06	100	92.51	100	95.24
RF	68.74	68.33	87.18	90.38	84.95	90.1	92.15	91.71	98.81	67.03	63.33	86.25	88.23	88.46	88.92	92.25	90.07	98.72
DT	83.47	88.33	90.26	98.57	88.76	92.95	94.57	88.86	98.8	82.93	87.03	89.8	97.39	85.2	91.21	89.72	84.61	98.59
LDA	**90.68**	**91.67**	98.33	100	97.14	100	**99.01**	98.57	100	**90.21**	**89.08**	98.22	100	96.15	100	**98.48**	98.06	100

TABLE 3.
Train-Test Accuracy and F_1-score.

	Accuracy									F_1-score								
	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian
KNN	64.1	66.67	92	96.55	96.55	100	95.12	96.55	96.08	63.89	66.43	91.32	96.48	96.48	100	92.35	96.49	95.76
LR	**79.49**	75	92	93.1	93.1	100	96.34	96.55	100	79.47	71.88	91.67	92.59	92.59	100	94.82	96.66	100
GNB	61.54	79.17	80	75.86	75.86	100	97.56	100	93.14	57.52	77.23	80	85.26	85.26	100	97.3	100	92.77
SVM-L	**76.92**	**83.33**	**92**	**96.55**	**96.55**	**100**	**96.34**	**100**	**100**	**76.92**	**82.22**	**91.67**	**96.48**	**96.48**	**100**	**94.82**	**100**	**100**
SVM-S	74.36	62.5	84	79.31	79.31	100	91.46	96.55	97.06	74.34	46.93	81.62	67.7	67.7	100	86.3	96.49	96.76
RF	61.54	50	80	75.86	75.86	93.1	86.59	89.66	100	60.61	39.5	76.19	85.26	87.95	89.49	77.7	89.65	100
DT	79.49	58.33	76	89.66	75.86	89.66	91.46	89.66	95.1	79.47	53.12	75	89.08	70.2	85.13	86.3	89.97	94.84
LDA	**76.92**	**79.17**	**96**	**100**	**100**	**100**	**98.78**	**93.1**	**100**	**76.86**	**75.76**	**95.76**	**100**	**100**	**100**	**98.7**	**93.33**	**100**

Considering Table 4, which reveals that the SVM-L model employing the Train-Validation-Test is superior to all other models, including the LDA, except in one case where (Breast cancer) dataset is used, where the LDA model is considered better than the other models tested with same type of datasets. However, KNN model has the highest Accuracy and F1-score of 77.78 and 75 respectively on the CNS dataset.

TABLE 4.
Train-Validation-Test Accuracy and F_1-score.

	Accuracy									F1-score								
	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian
KNN	46.67	**77.78**	60	81.82	81.82	90.91	87.1	72.73	94.74	40	**75**	58.33	85.71	85.71	87.06	74.64	73.89	92.72
LR	53.33	66.67	70	81.82	81.82	100	93.55	90.91	100	52.49	66.67	69.7	59.34	61.9	100	88.81	89.63	100
GNB	40	66.67	60	90.91	90.91	100	93.55	90.91	86.84	40	66.67	58.33	92.67	92.67	100	88.81	91.53	83.55
SVM-L	**53.33**	**55.56**	**80**	**90.91**	**90.91**	**100**	**93.55**	**90.91**	**100**	**52.49**	**55**	**80**	**96.67**	**96.67**	**100**	**88.81**	**89.63**	**100**
SVM-S	60	33.33	50	81.82	90.91	90.91	83.87	81.82	84.21	59.82	50	45.05	83.75	96.67	87.06	73.47	81.9	79.64
RF	60	33.33	60	90.91	90.91	100	90.32	81.82	94.74	59.82	50	60	92.86	96.67	100	80.76	80.56	93.21
DT	40	66.67	70	72.73	90.91	90.91	77.42	72.73	97.37	38.91	64.94	69.7	56.35	86.32	89.52	59.86	67.96	96.49
LDA	**66.67**	33.33	60	72.73	81.82	90.91	93.55	72.73	100	**66.06**	32.5	58.33	78.02	84.13	87.06	88.81	73.02	100

To further investigate the efficiency of the three data splitting techniques applied, we also used the Friedman Test to compare the results. Tables 5 and 6 provide the p-values and ranking results of the Friedman Test, respectively. Since F_1-score and Accuracy are utilized as evaluation metrics for checking the performance of the classifiers, the higher the Friedman Test ranking value, the better the performance (see Table 6).

From Table 6 in which each classifier has its own ranking scored using Friedman test; it can be noticed that SVM_L performance is the best with the three data separation techniques, followed by the LDA, which performed well with the TT and CV methods, while the LR performed well with the TVT method

TABLE 5.
Friedman Test P-values

Method	Accuracy	F_1-score
Train Test	0.00005238	0.00008898
Train Validation Test	0.03999788	0.03994592
5 Folds Cross Validation	0.00000658	0.00014103

.

TABLE 6

Classifier	Train Test		Train Validation Test		5-Folds Cross-Validation	
	Accuracy	F_1-score	Accuracy	F_1-score	Accuracy	F_1-score
KNN	4.83	4.67	3.50	3.61	4.28	4.28
LR	5.67	5.83	**5.56**	**5.22**	5.72	5.89
GNB	3.89	4.17	5.11	5.11	4.22	4.28
SVM_L	**6.39**	**6.50**	**6.28**	**6.44**	**6.83**	**6.56**
SVM_S	3.89	3.17	3.28	3.44	3.22	3.67
RF	2.17	2.44	4.83	5.17	1.78	2.00
DT	2.61	2.78	3.56	3.17	3.28	3.11
LDA	**6.56**	**6.44**	3.89	3.83	**6.67**	**6.22**

Friedman Test Ranking

Fig. 3, Fig. 4 and Fig. 5 visualize Accuracy and F_1-score ranking for TT, TVT and 5-Fold CV, respectively, depending on the results of Table 6. From the same figure (Fig. 6) it is clear that RF and GNB classifiers are less affected by the data separation technique than the other classifiers, while the LDA is the mostly affected.

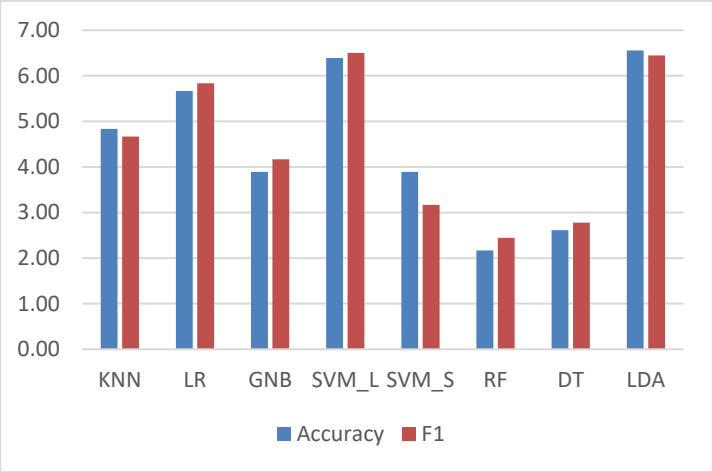


Fig. 3. Friedman Test Ranking for the Train Test method.

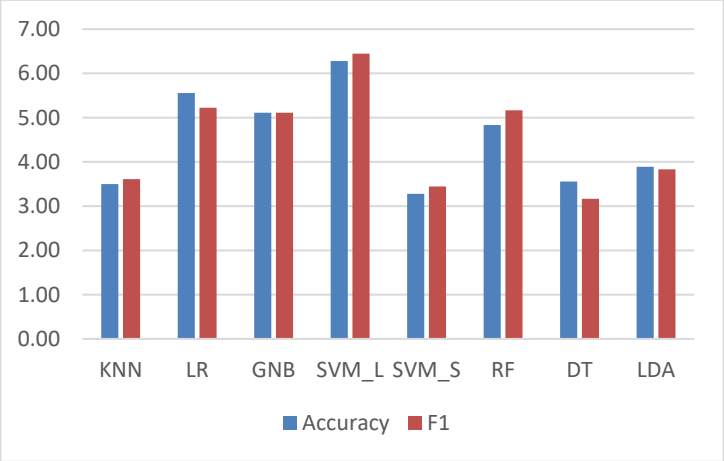


Fig. 4. Friedman Test Ranking for the Train Validation Test method.

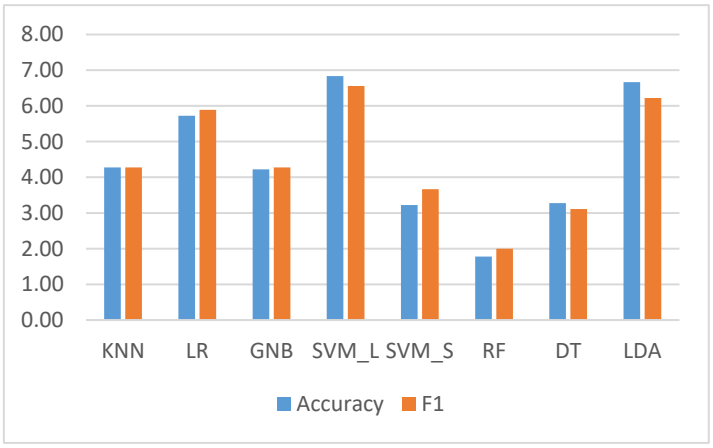


Fig. 5. Friedman Test Ranking for the 5-Folds Cross-Validation method.

Table 7. Standard Deviation of Accuracy and F_1-score for the three data separation methods

	Accuracy									F_1-score								
	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian	Breast	CNS	Colon	Leukemia_3c	Leukemia_4c	Leukemia	Lung	MLL	Ovarian
KNN	17.33	10.83	20.59	8.71	7.95	5.25	5.47	13.96	1.05	20.49	9.77	21.35	6.16	5.91	7.47	11.63	13.17	1.98
LR	17.55	11.82	13.69	9.18	7.94	0.00	2.04	3.97	0.00	17.34	10.59	13.50	21.66	19.31	0.00	4.04	4.69	0.00
GNB	12.80	8.67	14.14	12.19	10.43	0.00	2.31	5.25	4.55	10.21	6.25	14.86	7.37	6.26	0.00	4.37	4.89	6.15
SVM-L	18.87	18.91	10.07	4.59	3.47	0.00	2.27	4.89	0.00	19.21	15.98	8.25	1.98	0.63	0.00	4.54	5.66	0.00
SVM-S	9.64	17.61	20.61	9.65	8.47	5.25	5.10	9.66	7.05	9.55	17.55	20.95	14.40	16.84	7.47	9.71	9.60	9.48
RF	4.67	17.51	14.14	8.54	7.58	5.08	2.83	5.22	2.76	3.95	11.94	13.24	3.83	4.89	6.24	7.67	5.37	3.61
DT	24.03	15.48	10.41	13.13	8.14	1.66	9.14	9.55	1.87	24.48	17.21	10.42	21.70	9.00	3.14	16.34	11.48	1.88
LDA	12.05	30.72	21.49	15.74	9.78	5.25	3.09	13.62	0.00	12.10	29.58	22.35	12.69	8.28	7.47	5.65	13.30	0.00

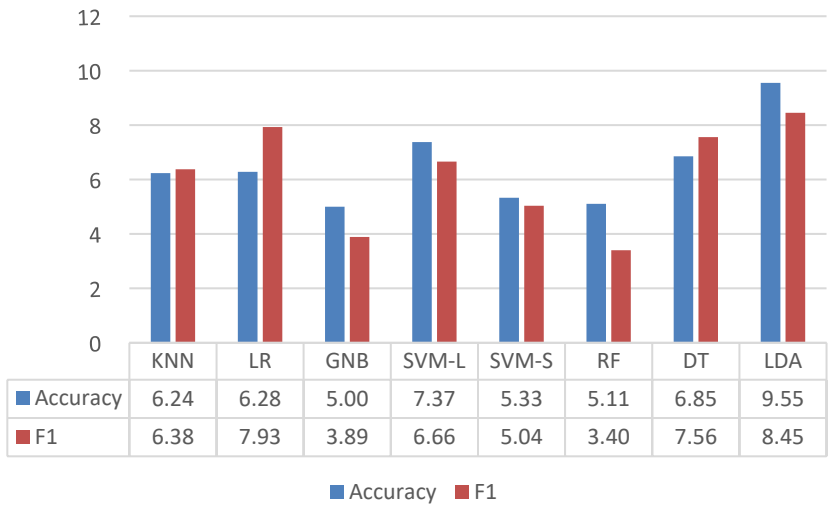


Fig. 6. Overall Standard Deviation of each algorithm.

The results in Fig. 6 have been calculated from Table 7 by taking the average of the Standard Deviations of the 9 datasets for each classifier according to their accuracy and F_1-score. Thus, obtaining the overall Standard deviation for each classifier of the 8 classifiers used in this research.

From our findings, we observed that overfitting is taking place in most of the classifiers investigated. This is due to the fact that the number of samples in the investigated datasets is very small, and therefore, the classifiers cannot generalize well for unseen data. As a recommendation for addressing this issue, further research should consider oversampling methods to augment the gene data, and hence, more accurate performance will be shown in the study conducted.

6. CONCLUSION

This article investigated the effectiveness of three data splitting methods (TT, TVT, CV) on the performance of eight classifiers (KNN, LR, GNB, SVM-L, SVM-S, RF, DT, and LDA) applied to nine cancer gene expression datasets. The CFA was combined with the previously stated classifiers to function as wrapper-based gene selection approaches. Accuracy and F1-score were used as measure metrics in this study, and the final findings were compared using the Fridman Test. Experimental results showed that the LDA classifier is the most affected one among all the other classifiers, while GNB and RF are the less affected ones. Furthermore, the findings indicated that, while LDA and SVM-L performed better than the other algorithms across all datasets, there is a significant performance difference when the LR classifier performance is considered. In most often datasets used, the results of all algorithms demonstrated that the train-test method of data separation is more accurate than the train-validation-test method, while the cross-validation method was superior to both.

REFERENCES

Abdu-Aljabar, Rana Dhia'a, and Osama A. Awad. 2021. "A Comparative Analysis Study of Lung Cancer Detection and Relapse Prediction Using XGBoost Classifier." *IOP Conference Series: Materials Science and Engineering* 1076(1): 012048.

Alanni, Russul, Jingyu Hou, Hasseeb Azzawi, and Yong Xiang. 2019. "A Novel Gene Selection Algorithm for Cancer Classification Using Microarray Datasets." *BMC Medical Genomics* 12(1): 1–12.

Alba, Enrique, José García-Nieto, Laetitia Jourdan, and El Ghazali Talbi. 2007. "Gene Selection in Cancer Classification Using PSO/SVM and GA/SVM Hybrid Algorithms." In *2007 IEEE Congress on Evolutionary Computation, CEC 2007*,.

Alshamlan, Hala, Ghada Badr, and Yousef Alohali. 2019. "Microarray Gene Selection and Cancer Classification Method Using Artificial Bee Colony and SVM Algorithms (ABC-SVM)." In *Lecture Notes in Electrical Engineering*,.

Arshak, Yousif, and Adel Eesa. 2018. "A New Dimensional Reduction Based on Cuttlefish Algorithm for Human Cancer Gene Expression." *ICOASE 2018 - International Conference on Advanced Science and Engineering*: 48–53.

Baliarsingh, Santos Kumar, Swati Vipsita, and Bodhisattva Dash. 2020. "A New Optimal Gene Selection Approach for Cancer Classification Using Enhanced Jaya-Based Forest Optimization Algorithm." *Neural Computing and Applications* 32(12): 8599–8616.

Begum, Shemim et al. 2018. "Gene Selection for Diagnosis of Cancer in Microarray Data Using Memetic Algorithm." In , 441–49.

Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. 2013. "A Review of Feature Selection Methods on Synthetic Data." *Knowledge and Information Systems* 34(3): 483–519.

Chen, Kun Huang, Kung Jeng Wang, Kung Min Wang, and Melani Adrian Angelia. 2014. "Applying Particle Swarm Optimization-Based Decision Tree Classifier for Cancer Classification on Gene Expression Data." *Applied Soft Computing Journal* 24: 773–80.

Dash, Rasmita, Rajashree Dash, and Rasmita Rautray. 2022. "An Evolutionary Framework Based Microarray Gene Selection and Classification Approach Using Binary Shuffled Frog Leaping Algorithm." *Journal of King Saud University - Computer and Information Sciences* 34(3): 880–91.

Dino, Hivi Ismat, Haval Ismael Hussein, Masoud Muhammed Hassan, and Adel Sabry Eesa. 2022. "Gene Expression Microarray Data Classification Based on PCA and Cuttlefish Algorithm." In *2022 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, 277–82.

Eesa, Adel Sabry, Zeynep Orman, and Adnan Mohsin Abdulazeez Brifcani. 2015. "A Novel Feature-Selection Approach Based on the Cuttlefish Optimization Algorithm for Intrusion Detection Systems." *Expert Systems with Applications* 42(5): 2670–79.

Fahrudin, Tresna Maulana, Iwan Syarif, and Ali Ridho Barakbah. 2017. "Ant Colony Algorithm for Feature Selection on Microarray Datasets." *Proceedings - 2016 International Electronics Symposium, IES 2016*: 351–56.

Guo, Shun, Donghui Guo, Lifei Chen, and Qingshan Jiang. 2016. "A Centroid-Based Gene Selection Method for Microarray Data Classification." *Journal of Theoretical Biology* 400: 32–41.

Jansi Rani, M., and D. Devaraj. 2019. "Two-Stage Hybrid Gene Selection Using Mutual Information and Genetic Algorithm for Cancer Data Classification." *Journal of Medical Systems* 43(8): 235.

Kar, Subhajit, Kaushik Das Sharma, and Madhubanti Maitra. 2015a. "Gene Selection from Microarray Gene Expression Data for Classification of Cancer Subgroups Employing PSO and Adaptive K-Nearest Neighborhood Technique." *Expert Systems with Applications* 42(1): 612–27.

Kar, Subhajit, Kaushik das Sharma, and Madhubanti Maitra. 2015b. "Gene Selection from Microarray Gene Expression Data for Classification of Cancer Subgroups Employing PSO and Adaptive K-Nearest Neighborhood Technique." *Expert Systems with Applications* 42(1).

Lee, Chien-Pang, and Yungho Leu. 2011. "A Novel Hybrid Feature Selection Method for Microarray Data Analysis." *Applied Soft Computing* 11(1): 208–13.

Li, Tao, Chengliang Zhang, and Mitsunori Ogihara. 2004. "A Comparative Study of Feature Selection and Multiclass Classfication Methods for Tissue Classification Based on Gene Expression." *Bioinformatics* 20(15).

Mohamad, Mohd Saberi, Sigeru Omatu, Safaai Deris, and Michifumi Yoshioka. 2009. "Particle Swarm Optimization for Gene Selection in Classifying Cancer Classes." *Artificial Life and Robotics* 14(1): 16–19.

Ooi, C. H., and Patrick Tan. 2003. "Genetic Algorithms Applied to Multi-Class Prediction for the Analysis of Gene Expression Data." *Bioinformatics* 19(1).

Othman, Mohd Shahizan, Shamini Raja Kumaran, and Lizawati Mi Yusuf. 2020. "Gene Selection Using Hybrid Multi-Objective Cuckoo Search Algorithm With Evolutionary Operators for Cancer Microarray Data." *IEEE Access* 8: 186348–61.

Pragadeesh, C. et al. 2019. "Hybrid Feature Selection Using Micro Genetic Algorithm on Microarray Gene Expression Data." *Journal of Intelligent & Fuzzy Systems* 36(3): 2241–46.

Sabry Eesa, Adel, Adnan Mohsin Abdulazeez, and Zeynep Orman. 2013. "Cuttlefish Algorithm – A Novel Bio-Inspired." *International Journal of Scientific & Engineering Research* 4(9): 1978–86.

Sahu, Barnali, and Debahuti Mishra. 2012. "A Novel Feature Selection Algorithm Using Particle Swarm Optimization for Cancer Microarray Data." *Procedia Engineering* 38: 27–31.

Sayed, Sabah, Mohammad Nassef, Amr Badr, and Ibrahim Farag. 2019. "A Nested Genetic Algorithm for Feature Selection in High-Dimensional Cancer Microarray Datasets." *Expert Systems with Applications* 121: 233–43.
Settouti, Nesma, Mohammed El Amine Bechar, and Mohammed Amine Chikh. 2016. "Statistical Comparisons of the Top 10 Algorithms in Data Mining for Classi Cation Task." *International Journal of Interactive Multimedia and Artificial Intelligence* 4(1): 46.
Shi, Zhiao, Bo Wen, Qiang Gao, and Bing Zhang. 2021. "Feature Selection Methods for Protein Biomarker Discovery from Proteomics or Multiomics Data." *Molecular and Cellular Proteomics* 20: 100083.
Soufan, Othman, Dimitrios Kleftogiannis, Panos Kalnis, and Vladimir B. Bajic. 2015. "DWFS: A Wrapper Feature Selection Tool Based on a Parallel Genetic Algorithm." *PLOS ONE* 10(2): e0117988.
Tabakhi, Sina, Ali Najafi, Reza Ranjbar, and Parham Moradi. 2015. "Gene Selection for Microarray Data Classification Using a Novel Ant Colony Optimization." *Neurocomputing* 168: 1024–36.
Vafaee, Fatemeh, Sara Mosafer, and Mohammad Hossein. 2016a. "Genomics A Hybrid Gene Selection Approach for Microarray Data Classi Fi Cation Using Cellular Learning Automata and Ant Colony Optimization." *Genomics* 107(6): 231–38.
Wang, Aiguo et al. 2017. "Wrapper-Based Gene Selection with Markov Blanket." *Computers in Biology and Medicine* 81(September 2016): 11–23.
Zhu, Zexuan, Yew-Soon Ong, and Manoranjan Dash. 2007. "Markov Blanket-Embedded Genetic Algorithm for Gene Selection." *Pattern Recognition* 40(11): 3236–48.