Journal of Mathematical Modeling
Vol. 13, No. 2, 2025, pp. 341–356. Research Article

JMM

Feature selection via mixed-integer program and supervised infinite feature selection method

Mohammad Noroozi[†], **Maziar Salahi**[*†], **Sadegh Eskandari**[‡]

[†]*Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran*

[‡]*Department of Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran*

Email(s): mhmdnrz91@gmail.com, salahim@guilan.ac.ir, eskandari@guilan.ac.ir

Abstract. Feature selection is an important step in data preprocessing, which helps reducing the dimensionality of data and simplifying the models. This process not only reduces the computational complexity of models, but also improves their accuracy by eliminating irrelevant features and noise. The three most widely used approaches for feature selection are filter, wrapper and embedded methods. In this paper, first we review some support vector machine based Mixed-Integer Linear Programming (MILP) models and Supervised Infinite Feature Selection (Inf-FS$_s$) method. Then, we propose three hybrid approaches based on them. The first approach involves solving the relaxed linear model of the underlying MILP model and then solving the MILP model for those features with nonzero weights, namely a smaller MILP. In the second approach, first the Inf-FS$_s$ method is applied to rank the features. Then depending on the features costs, either chooses the top features from the ranked features until budget parameter is reached or solves a knapsack problem to select cost effective features. The third approach applies the first approach to the top 20% of features ranked by Inf-FS$_s$ method. To evaluate the proposed approaches' performance, experiments are conducted on four high-dimensional benchmark datasets for fixed and random features costs. Results demonstrate that using either of the proposed approaches can significantly reduce running time of MILP models with comparable accuracies with the original MILP models.

Keywords: Feature selection, Mixed integer linear program, Infinite feature selection method, Feature cost.
AMS Subject Classification : 68T09, 90C11.

1 Introduction

In todays rapidly evolving data landscape, feature selection plays a pivotal role in data analysis and machine learning. The primary goal of feature selection is to reduce the dimensionality of the data, enhancing model performance by retaining only the most relevant and significant features [12]. This

*Corresponding author
Received: 20 October 2024 / Revised: 26 December 2024 / Accepted: 29 December 2024
DOI: 10.22124/jmm.2024.28684.2557

process not only mitigates computational costs but also improves model accuracy and prevents overfitting. Especially in problems with a small number of samples and high dimensionality, the performance of the model significantly improves by using a small subset of features [5, 14].

Guyon et al. [9] classified feature selection techniques into three primary categories: filter methods, wrapper methods and embedded methods. Filter methods exclude poorly informative features by analyzing their statistical properties before applying a classification algorithm. Wrapper methods work alongside the learning model, examining the entire set of variables to find feature subsets based on their predictive performance. While this approach is computationally intensive, it often yields better results than filter methods. Embedded methods, belonging to the third category, perform feature selection while simultaneously building the classifier. This can be viewed as a search through both feature subsets and hypothesis spaces. Unlike wrapper methods, which rely on a separate classification algorithm, embedded methods handle both tasks within a single process. Typically, they are less computationally demanding than wrapper methods [9].

Despite the progress in feature selection techniques [4, 8], there remains a significant challenge in addressing real-world constraints such as varied feature costs and limited budgets. Cost considerations are essential for making sensible and effective decisions in many real-world applications. Ignoring the cost factor in the feature selection process can lead to significantly undesirable outcomes. By incorporating a budget constraint, the process ensures the selection of the most informative yet cost-effective features, while preserving classification accuracy within a predefined budget. This makes the feature selection process more practical and efficient for real-world use [12]. Support Vector Machine (SVM) based feature selection approaches incorporate cost-effective feature selection in the original SVM-based models using zero-one variables. For example, Maldonado et al. [13] introduced two Mixed-Integer Linear Programming (MILP) models based on the L_1-SVM model [3] and the LP-SVM model [16], called MILP1 and MILP2, respectively, to address feature selection with budget constraints. Then, Labbe et al. [11], based on the idea introduced by Maldonado et al., proposed an extension of L_1-SVM and applied two strategies to enhance bounds on the zero-one variables. Another extension of L_1-SVM based feature selection was proposed by Lee et al. [12] called group feature selection. They further studied its robust counterpart under uncertainty.

An issue with MILP models is that as the dimensionality increases, the time required to solve these models grows significantly. Therefore, proposing methods that reduce computational time without reducing accuracy becomes crucial. In this paper, we propose three hybrid approaches, two of which are aimed at reducing the computational time of MILP models while maintaining high accuracy. In the first approach, firstly the linear programming relaxation of the MILP model is solved. Then those features with nonzero weights are chosen and the original MILP model is applied with these less number of features. This approach is particularly advantageous for high-dimensional data, offering a reduction in computational burden without substantially impacting model accuracy. In the second approach, firstly the supervised Infinite Feature Selection (Inf-FS$_s$) [15] is applied to rank the features using three criterion, namely Fisher score, mutual information and standard deviation. As Inf-FS$_s$ do not take into account features' costs, we either choose the first B features, where B is the budget on the features' costs or we solve a knapsack problem to select cost effective features. The third approach applies the first approach to the MILP models, restricted to the top 20% of features ranked by Inf-FS$_s$ method.

The remainder of the paper is structured as follows. In Section 2, we review two MILP feature selection models with budget constraint and the Inf-FS$_s$ method. In Section 3, we introduce the proposed approaches. Finally, in Section 4, we compare the computational time and accuracy of the proposed approaches with the MILP models discussed in Section 2.

2 MILP models and Inf-FS$_s$ method for feature selection

In this section, first we present two embedded-based MILP models for feature selection with the budget constraint. Then we present the filter-based Inf-FS$_s$ method that ranks features based on Fisher score, mutual information and standard deviation while not taking into account features' costs.

2.1 MILP models

The first MILP model, which we call MILP1, is introduced by Maldonado et al. [13] as follows:

$$\begin{aligned}
\min_{w,v,b,\varepsilon} \quad & \sum_{i=1}^{m} \varepsilon_i \\
\text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \quad i = 1,\ldots,m, \\
& l_j v_j \leq w_j \leq u_j v_j, \quad j = 1,\ldots,n, \\
& \sum_{j=1}^{n} c_j v_j \leq B, \\
& v_j \in \{0,1\}, \quad j = 1,\ldots,n, \\
& \varepsilon_i \geq 0, \quad i = 1,\ldots,m,
\end{aligned} \tag{1}$$

where c_j is the cost of the jth feature, the second set of constraints bounds the features' weights, and the third constraint ensures that the total cost of the selected features does not exceed the predefined budget, B. If all c_js are set to one, the budget parameter B represents the maximum number of selected features. If $v_j = 0$, then from the second set of constraints $w_j = 0$ that means jth feature is not selected. Otherwise its weight is bounded below by l_j and above by u_j.

The second MILP model, which we call MILP2, is introduced by Labbe et al. [11] as follows:

$$\begin{aligned}
\min_{v,w^+,w^-,b,\varepsilon} \quad & \sum_{j=1}^{n} (w_j^+ + w_j^-) + C \sum_{i=1}^{m} \varepsilon_i \\
\text{s.t.} \quad & y_i \left(\sum_{j=1}^{n} (w_j^+ - w_j^-) x_{ij} + b \right) \geq 1 - \varepsilon_i, \quad i = 1,\ldots,m, \\
& w_j^+ \leq u_j v_j, \quad j = 1,\ldots,n, \\
& w_j^- \leq -l_j v_j, \quad j = 1,\ldots,n, \\
& \sum_{j=1}^{n} c_j v_j \leq B, \\
& v_j \in \{0,1\}, \quad j = 1,\ldots,n, \\
& \varepsilon_i \geq 0, \quad i = 1,\ldots,m, \\
& w_j^+ \geq 0, \quad i = 1,\ldots,n, \\
& w_j^- \geq 0, \quad i = 1,\ldots,n,
\end{aligned} \tag{2}$$

where the parameter C in the objective function reflects the penalty for errors. It uses the decomposition of w into w^+ and w^- for linearization. This formulation seeks an optimal balance between deviations and the margin. As can be observed, the second and third sets of constraints bound the features' weights

and the fourth constraint is the budget constraint on features. Similar to MILP1, If $v_j = 0$, then from the second and third sets of constraints $w_j = 0$ that means j-th feature is not selected. Their experiments showed that choosing very conservative big values for u_j and l_j for all $j = 1,\ldots,n$ might lead to high computational costs. Therefore, they proposed two strategies for tightening u_j and l_j values to reach better computational results [11].

2.2 Inf-FS$_s$ method

Inf-FS$_s$ is a filter-based feature selection models, which was introduced by G. Roffo et al. [15]. This method uses three criteria to calculate the score of each feature: Fisher score (h_i), mutual information (m_i), and standard deviation (σ_i). In this approach, a complete graph $G = (V,E)$ is constructed where nodes represent features and edges indicate the relationship between them. Each edge $A(i,j)$ shows the probability of selecting features v_i and v_j. The Inf-FS$_s$ algorithm works by selecting features from a set of features $F = \{f_1,\ldots,f_n\}$, where G represents the set of classes (labels) and $Y \in \{1,\ldots,G\}$ are the class labels. The algorithm proceeds as follows:

Algorithm 1 Supervised Infinite Feature Selection Algorithm

Input: Set of features $F = \{f_1,\ldots,f_n\}$, class labels Y, and coefficients $\alpha_1,\alpha_2,\alpha_3$.
Output: Final ranking $\hat{C}$ for each feature.

1. For each $i = 1,\ldots,n$
Compute:

$$h_i = \frac{|\mu_{i,1} - \mu_{i,2}|^2}{\sigma_{i,1}^2 + \sigma_{i,2}^2}, \quad m_i = \sum_{y\in Y}\sum_{z\in F_i} p(z,y)\log\left(\frac{p(z,y)}{p(z)p(y)}\right)$$

where $\mu_{i,1},\mu_{i,2}$ are the means for each class, and $\sigma_{i,1},\sigma_{i,2}$ are the standard deviations for each class.
Compute the score for each feature as:

$$s_i = \alpha_1 h_i + \alpha_2 m_i + \alpha_3 \sigma_i.$$

2. For each $i = 1,\ldots,n$ and $j = 1,\ldots,n$
Set:

$$A(i,j) = s_i s_j.$$

3. Compute r as:

$$r = \frac{0.9}{\rho(A)},$$

where $\rho(A)$ is the spectral radius of matrix A.
4. Compute the final score $\hat{C}$ as:

$$\hat{C} = \left((I - rA)^{-1} - I\right)e,$$

where I is the identity matrix and e is the vector of ones.

Note that the coefficients α_1, α_2 and α_3 take values between 0 and 1 with sum equals 1 that are determined using cross-validation.

3 Proposed approaches

In this section, we introduce simple hybrid approaches for the feature selection problem taking into account budget constraints on the features' costs.

- **Approach 1:** Consider MILP1 model (1). First we solve its relaxed model as follows:

$$\begin{aligned}
\min_{w,v,b,\varepsilon} \quad & \sum_{i=1}^{m} \varepsilon_i \\
\text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \quad i = 1,\ldots,m, \\
& l_j v_j \leq w_j \leq u_j v_j, \quad j = 1,\ldots,n, \\
& \sum_{j=1}^{n} c_j v_j \leq B, \\
& 0 \leq v_j \leq 1, \quad j = 1,\ldots,n, \\
& \varepsilon_i \geq 0, \quad i = 1,\ldots,m.
\end{aligned} \tag{3}$$

  The solution to this problem is typically sparse. Therefore, we choose features with nonzero weights (for instance $|w_j| > 10^{-6}$) and solve MILP1 model only for these features, namely an MILP model with less zero-one variables. Let $P = \{j||w_j| > 10^{-6}\}$, then the following MILP is solved:

$$\begin{aligned}
\min_{w,v,b,\varepsilon} \quad & \sum_{i=1}^{m} \varepsilon_i \\
\text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \quad i = 1,\ldots,m, \\
& l_j v_j \leq w_j \leq u_j v_j, \quad j \in P \\
& \sum_{j\in P} c_j v_j \leq B, \\
& v_j \in \{0,1\}, \quad j \in P \\
& \varepsilon_i \geq 0, \quad i = 1,\ldots,m.
\end{aligned} \tag{4}$$

  Since the cardinality of P is much smaller than n, MILP (4) is usually solved faster than the original MILP1. Similarly, we can apply this approach to the MILP2.

- **Approach 2:** First, using Inf-FS$_s$ method features are ranked. If costs are all equal to one, then we just choose the first B features from the ranked features. However, if features costs are different, we solve the following knapsack problem to select cost effective features:

$$\begin{aligned}
\max_{x} \quad & \sum_{i=1}^{n} s_i x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} c_i x_i \leq B \\
& x_i \in \{0,1\}, \quad i = 1,\ldots,n
\end{aligned} \tag{5}$$

  where s_i and c_i are the score and cost of the ith feature, respectively.

- **Approach 3:** In this approach, we first apply Inf-FS$_s$ to rank the features. Then we apply Approach 1 to the original MILP models, restricted to the top 20% of features ranked by Inf-FS$_s$ method.

4 Experimental results

In this section, the proposed approaches in Section 3 are evaluated on four datasets and their results are compared with the original MILP models and Inf-FS$_s$ method in terms of CPU time and accuracy. All implementations are carried out on MATLAB. The optimization problems are formulated using CVX and solved using the CPLEX solver [6]. Table 1 shows the abbreviations in this section.

Table 1: List of abbreviations.

Abbreviation	Description
MILP1*	Application of Approach 1 to MILP1
MILP2*	Application of Approach 1 to MILP2
MILP1$^*_s$	Application of Approach 3 to MILP1
MILP2$^*_s$	Application of Approach 3 to MILP2
K-Inf-FS$_s$	Model (5)
B	Budget parameter
$F_C \cap F_D$	The intersection of the selected features by approaches C and D

Furthermore, in order to evaluate the accuracy, a 10-fold cross-validation method is used, where the dataset is split into 10 equal folds [5]. The model is trained on nine of these folds, while the remaining fold (test set) is used to evaluate the model's prediction error. This procedure is repeated for all 10 folds. Accuracy is calculated as the ratio of correctly classified samples to the total number of samples in the testing set and is defined as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}.$$

In this formula,

- TP (True Positives) refers to the number of positive samples correctly classified,
- TN (True Negatives) refers to the number of negative samples correctly classified,
- FP (False Positives) refers to the number of negative samples incorrectly classified as positive, and
- FN (False Negatives) refers to the number of positive samples incorrectly classified as negative.

4.1 Datasets and parameters setting

Table 2 shows the datasets information, where m and n represent the number of samples and the number of features, respectively. Using an appropriate penalty parameter affects the accuracy of the model. For each dataset, penalty parameters from the $\{2^{-7}, 2^{-6}, \ldots, 2^{-1}, 2^0, 2^1, \ldots, 2^6, 2^7\}$ are considered, and the best one is selected based on accuracy that are given in Table 3. It should be noted that the MILP1 model does not have a penalty parameter. Furthermore, the α parameters (α_1, α_2, and α_3) that are most suitable for the Inf-FS$_s$ model are presented in Table 4. These values are determined through cross-validation.

4.2 Equal features costs

In the first scenario, the cost of each feature is considered fixed and equal to one. Moreover, to evaluate the model's performance under different budget parameter, the values 10, 20, and 30 are assigned to the

Table 2: Datasets information.

DATASETS	$m \times n$
COLONCANCER [1]	62×2000
LYMPHOMA [7]	45×4026
LEUKEMIA [7]	72×7070
PROSTATE [2]	102×6033

Table 3: The best penalty parameter (C) used for the datasets in each model.

DATASET	MILP1	MILP2	Inf-FS$_s$
COLONCANCER	-	2^0	2^{-4}
LYMPHOMA	-	2^5	2^0
LEUKEMIA	-	2^5	2^7
PROSTATE	-	2^{-1}	2^{-7}

Table 4: The α parameters in the Inf-FS$_s$ method.

DATASET	COLONCANCER	LYMPHOMA	LEUKEMIA	PROSTATE
α_1	0.8	0.5	0.9	0.6
α_2	0.1	0.2	0	0.1
α_3	0.1	0.3	0.1	0.3

budget parameter (B). The best results in terms of accuracy and execution time for each budget parameter are bold numbers in the tables and the second best result is marked with a superscript "*".

Table 5: Time and accuracy comparison for COLONCANCER dataset.

Model		$B = 10$	$B = 20$	$B = 30$
MILP1	Accuracy	88.6	88.8	85.7
	Time	11.2	10000	10000
MILP1*	Accuracy	88.8*	88.3	85.1
	Time	9.7*	860	10000
MILP2	Accuracy	88.6	**90.5**	87.6*
	Time	21.3	10000	10000
MILP2*	Accuracy	88.8*	**90.5**	86.7
	Time	10	942	10000
Inf-FS$_s$	Accuracy	87.5	80	82.2
	Time	**7.8**	**7.9**	**7.8**
MILP1$_s^*$	Accuracy	**90**	88.6	87.1
	Time	12.8	57.5*	1920.7*
MILP2$_s^*$	Accuracy	88.8*	89	**88.6**
	Time	12.6	67	2137.7

For the COLONCANCER dataset, one can see from Table 5 that the Inf-FS$_s$ model is the fastest in terms of execution time across all budget scenarios. For $B = 10$, Approach 1 reduces the running times of MILP1 and MILP2 models. For $B = 20$, Approaches 1 and 3 both significantly reduces the running times of the models. In the case of $B = 30$, just Approach 3 reduces the running times of the original MILP models. From the accuracy point of view, Inf-FS$_s$ exhibits the worst accuracy among all models and we see no significant differences between the MILP models and the proposed approaches. In the cases of $B = 10$ and $B = 30$, MILP1$_s^*$ and MILP2$_s^*$ have the highest accuracy, while for the case $B = 20$, MILP2 and MILP2* have jointly the highest accuracy.

Table 6: Features overlap percentage on COLONCANCER dataset.

B	10	20	30
$F_{MILP1} \cap F_{MILP1^*}$	77.3	**83.6**	72.4
$F_{MILP2} \cap F_{MILP2^*}$	**100**	74	**81.9**
$F_{MILP1^*} \cap F_{MILP2^*}$	72.7	80*	75.5
$F_{MILP1^*} \cap F_{Inf\text{-}FS_s}$	80	60	50
$F_{MILP2^*} \cap F_{Inf\text{-}FS_s}$	**100**	50	40
$F_{MILP1^*_s} \cap F_{MILP1}$	72.8	76.5	66.5
$F_{MILP2^*_s} \cap F_{MILP2}$	92.2*	68.2	74.6
$F_{MILP1^*_s} \cap F_{MILP2^*_s}$	72.7	80*	77.1*
$F_{MILP1^*_s} \cap F_{Inf\text{-}FS_s}$	70	50	45
$F_{MILP2^*_s} \cap F_{Inf\text{-}FS_s}$	70	40	42

We also report features' overlap between all methods in Table 6 for the COLONCANCER dataset. As we see, the proposed Approaches 1 and 3 have high overlap with the original MILP models. Furthermore, in cases of $B = 20$ and 30 one can see that features overlap between MILP models and Inf-FS$_s$ is the worst, that might be due to the fact that they belong to different features selection methods (embedded and filter methods). Moreover, as expected MILP models have more features in common with Approach 1 than with Approach 3.

Table 7: Time and accuracy comparison for LYMPHOMA dataset.

Model		$B = 10$	$B = 20$	$B = 30$
MILP1	Accuracy	93.3	91.5	91.7
	Time	51	10000	60
MILP1*	Accuracy	93.3	93.3*	95.8*
	Time	26*	66.8	25.8
MILP2	Accuracy	94.5*	91.8	89.5
	Time	70	10000	18.3*
MILP2*	Accuracy	93.8	92.5	90
	Time	**20.8**	**37.3**	**7.2**
Inf-FS$_s$	Accuracy	**95.8**	**100**	**98.5**
	Time	40.3	45*	45
MILP1$^*_s$	Accuracy	93.5	92	95.5
	Time	62.2	74	63.1
MILP2$^*_s$	Accuracy	92.5	91.8	90.5
	Time	61.4	70.2	62

The results for the LYMPHOMA dataset are reported in Tables 7 and 8. The reduction in execution time in the Approach 1 compared to the original MILP models is evident for all budget scenarios. Approach 3 for the case of $B = 10$ reduces the running time of MILP2. In case of $B = 20$, Approach 3 reduces the running times of MILP models. From the accuracy point of view, Inf-FS$_s$ has the best accuracy for all budget scenarios.

According to the features' overlap between all methods in Table 8 for the LYMPHOMA dataset, one

Table 8: Features overlap percentage on LYMPHOMA dataset.

B	10	20	30
$F_{MILP1} \cap F_{MILP1^*}$	89.5	78.8	61.8
$F_{MILP2} \cap F_{MILP2^*}$	**94.7**	**91.1**	**81**
$F_{MILP1^*} \cap F_{MILP2^*}$	84.2	84.6	63.1
$F_{MILP1^*} \cap F_{Inf\text{-}FS_s}$	50	50	56.6
$F_{MILP2^*} \cap F_{Inf\text{-}FS_s}$	40	50	46.6
$F_{MILP1_s^*} \cap F_{MILP1}$	84.2	73.6	56
$F_{MILP2_s^*} \cap F_{MILP2}$	89.1	86.6	76.6
$F_{MILP1_s^*} \cap F_{MILP2_s^*}$	94.4*	86.8*	71.2*
$F_{MILP1_s^*} \cap F_{Inf\text{-}FS_s}$	70	50	70
$F_{MILP2_s^*} \cap F_{Inf\text{-}FS_s}$	60	50	63.3

can see that the proposed approaches have high overlap with the original MILP models.

Table 9: Time and accuracy comparison for LEUKEMIA dataset.

Model		$B = 10$	$B = 20$	$B = 30$
MILP1	Accuracy	96.9	94.6	97.3
	Time	6820	252	119
MILP1*	Accuracy	95.2	98.5*	96.8
	Time	**40**	**25.8**	**23.8**
MILP2	Accuracy	**97.6**	97.3	97
	Time	10000	10000	2312
MILP2*	Accuracy	96.1	**98.8**	97
	Time	49.1*	1973	38.1*
Inf-FS$_s$	Accuracy	97.3*	**98.8**	**100**
	Time	137	150*	150.5
MILP1$_s^*$	Accuracy	96.3	97.2	97.8*
	Time	197.5	193.8	190.2
MILP2$_s^*$	Accuracy	97.1	**98.8**	94.4
	Time	200.5	356	198

The results for LEUKEMIA dataset are summarized in Tables 9 and 10. The reduction in execution times in the Approaches 1 and 3 compared to the original MILP models is evident. According to the Table 9, the best running times for all budget scenarios correspond to the MILP1* model. From the accuracy point of view, MILP2 has the best accuracy for the case $B = 10$, while in the case of $B = 20$ three approaches jointly have the best accuracy. As can be seen for the case of $B = 30$, Inf-FS$_s$ has the best accuracy.

Features overlap between all methods in Table 10 for the LEUKEMIA dataset show that the proposed Approaches 1 and 3 have high overlap with the original MILP models in most cases.

The results for PROSTATE dataset are reported in Tables 11 and 12. Except the case $B = 10$, for the other two budgets Inf-FS$_s$ has the worst accuracy. The best accuracy for the first two budget scenarios correspond to MILP1* and for the case $B = 30$ belongs to MILP1. Also, one can observe that the

Table 10: Features overlap percentage on LEUKEMIA dataset.

B	10	20	30
$F_{MILP1} \cap F_{MILP1^*}$	88.5	55	45
$F_{MILP2} \cap F_{MILP2^*}$	87.5	86.4*	96*
$F_{MILP1^*} \cap F_{MILP2^*}$	88.5	58.5	53
$F_{MILP1^*} \cap F_{Inf\text{-}FS_s}$	90*	60	66.6
$F_{MILP2^*} \cap F_{Inf\text{-}FS_s}$	**100**	60	66.6
$F_{MILP1^*_s} \cap F_{MILP1}$	76.4	46	39.6
$F_{MILP2^*_s} \cap F_{MILP2}$	76.6	73.3	85.2
$F_{MILP1^*_s} \cap F_{MILP2^*_s}$	87	**95.5**	**99**
$F_{MILP1^*_s} \cap F_{Inf\text{-}FS_s}$	70	70	56.7
$F_{MILP2^*_s} \cap F_{Inf\text{-}FS_s}$	80	70	63.3

Table 11: Time and accuracy comparison for PROSTATE dataset.

Model		$B = 10$	$B = 20$	$B = 30$
MILP1	Accuracy	90.6	91.2	**94.1**
	Time	197.2	351.6	10000
MILP1*	Accuracy	**91.2**	**92.5**	93.4*
	Time	102	118.5	767.5
MILP2	Accuracy	90.7*	91.2	93.1
	Time	137.4	184.7	10000
MILP2*	Accuracy	90.7*	91.2	93.1
	Time	**30.3**	**34**	593
Inf-FS$_s$	Accuracy	90.3	86	87.2
	Time	84*	85*	**84.2**
MILP1$^*_s$	Accuracy	90.3	92.3*	93.4*
	Time	113.2	120.6	136.2*
MILP2$^*_s$	Accuracy	90.2	91.2	93.2
	Time	112.6	125.8	139.1

proposed Approaches 1 and 3 perform faster compared to the original MILP models. The Inf-FS$_s$ is the fastest for $B = 30$ in Table 11. According to the Table 11, the best running times for the first two budget parameters correspond to MILP2*.

Finally, as before for the PROSTATE dataset one can see that the proposed Approaches 1 and 3 have again high overlap with the original MILP models.

4.3 Random features costs

In the second scenario, the cost of selecting each feature is randomly determined between 1 and 10. To evaluate the models' performance under different budget conditions, the values of 10, 20, 30, and 50 are considered for the budget parameter (B). Computational results are summarized in Tables 13-20.

The results for COLONCANCER dataset are presented in Table 13. As can be seen, the reduction in execution times in the Approaches 1 and 3 compared to the original MILP models is evident for all

Table 12: Features overlap percentage on PROSTATE dataset.

B	10	20	30
$F_{MILP1} \cap F_{MILP1^*}$	93.3	84.6	82.1
$F_{MILP2} \cap F_{MILP2^*}$	**100**	**97.4**	**94.9**
$F_{MILP1^*} \cap F_{MILP2^*}$	93.3*	86.8	82.1
$F_{MILP1^*} \cap F_{Inf\text{-}FS_s}$	60	50	43.3
$F_{MILP2^*} \cap F_{Inf\text{-}FS_s}$	60	50	50
$F_{MILP1^*_s} \cap F_{MILP1}$	91.2	82.6	78.3
$F_{MILP2^*_s} \cap F_{MILP2}$	97.8*	93.2*	87.4*
$F_{MILP1^*_s} \cap F_{MILP2^*_s}$	**100**	88.2	84.6
$F_{MILP1^*_s} \cap F_{Inf\text{-}FS_s}$	70	65	40
$F_{MILP2^*_s} \cap F_{Inf\text{-}FS_s}$	70	65	40

Table 13: Time and accuracy comparison for COLONCANCER dataset.

Model		$B=10$	$B=20$	$B=30$	$B=50$
MILP1	Accuracy	84.1	87.1	88.3*	86.6
	Time	17.1	19.5	239	10000
MILP1*	Accuracy	84	86.7	87.7	86.9
	Time	**9.5**	**11.8**	109	6580
MILP2	Accuracy	84	87	**88.4**	87.6
	Time	24.5	28.4	252	10000
MILP2*	Accuracy	84.3*	87	**88.4**	87.7
	Time	15.2	16.1	94	7610
K-Inf-FS$_s$	Accuracy	**87.8**	87	86.1	**90.4**
	Time	12.5*	13.1*	**13.2**	**13.7**
MILP1$^*_s$	Accuracy	83.5	87.8*	88.2	88.7
	Time	12.8	13.6	15.1*	49.4*
MILP2$^*_s$	Accuracy	83.8	**88.6**	88.2	89.2*
	Time	12.5*	13.4	15.3	55.3

budget scenarios. For $B = 10$ and 20, MILP1* is the fastest in terms of execution time. For $B = 30$ and 50, K-Inf-FS$_s$ has the best execution time. From the accuracy point of view, for $B = 10, 50$, K-Inf-FS$_s$ is the best, for $B = 20$ MILP2$^*_s$ is the best and for $B = 30$ MILP2 and MILP2* are jointly the bests. In general, for the budget parameter $B = 10$, the accuracies of the models all less than the other budget parameters.

From Table 14 for the COLONCANCER dataset, one can observe that the proposed Approaches 1 and 3 exhibit a high overlap with the original MILP models, similar to the case with fixed features costs. Also, it can be noted that features' overlap between between different methods (embedded and filter methods) is smaller compared to the case where two method belong to one class.

The results for LYMPHOMA dataset are presented in Tables 15 and 16. One can observe that the Approach 1 for all budget parameters performs faster than original MILP models. In addition, Approach 2 for budget cases $B = 30$ and 50 performs faster than original MILP models, while for the budget parameters $B = 10$ and 20 it doesnt. According to Table 15, the best running times, except for the budget

Table 14: Features overlap percentage on COLONCANCER dataset.

B	10	20	30	50
$F_{MILP1} \cap F_{MILP1^*}$	88.8	87.1	86	82.6
$F_{MILP2} \cap F_{MILP2^*}$	**97.7**	**100**	**98.4**	**92**
$F_{MILP1^*} \cap F_{MILP2^*}$	87.5	86.7	83.7	78.9
$F_{MILP1^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	63.6	61.5	48.2	61.9
$F_{MILP2^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	63.6	61.5	48.2	47.6
$F_{MILP1_s^*} \cap F_{MILP1}$	79.8	79.6	78.6	74
$F_{MILP2_s^*} \cap F_{MILP2}$	89.2	94.5	93.2*	85.5
$F_{MILP1_s^*} \cap F_{MILP2_s^*}$	92.2*	92.1*	89	88*
$F_{MILP1_s^*} \cap F_{Inf\text{-}FS_s}$	70	77.8	72.8	70.5
$F_{MILP2_s^*} \cap F_{Inf\text{-}FS_s}$	67.5	77.8	71.8	63.5

Table 15: Time and accuracy comparison for LYMPHOMA dataset.

Model		$B=10$	$B=20$	$B=30$	$B=50$
MILP1	Accuracy	91.5*	91	94	93.4
	Time	23.5	31.4	88.2	10000
$MILP1^*$	Accuracy	90.5	90.3	92	93.5
	Time	15*	19.8*	22.4*	221.5
MILP2	Accuracy	91.2	91.1	93.6	93.5
	Time	24.5	37.1	106	10000
$MILP2^*$	Accuracy	90.7	92*	94.4*	93.5
	Time	**12.8**	**15.4**	**21.3**	206
$K\text{-}Inf\text{-}FS_s$	Accuracy	**97.5**	**95.6**	**96.2**	**99.2**
	Time	44	44.5	45.2	**45.6**
$MILP1_s^*$	Accuracy	90.5	90.6	93.2	94.5
	Time	62.2	63	63.9	70.2*
$MILP2_s^*$	Accuracy	90	91	94	95.4*
	Time	61.7	62.4	62.3	76.6

parameter $B = 50$, correspond to $MILP2^*$ and for the case $B = 50$, $K\text{-}Inf\text{-}FS_s$ has the best running time. Also, the second best running times for all parameter budgets (except $B = 50$) belong to $MILP1^*$. In terms of accuracy, $K\text{-}Inf\text{-}FS_s$ exhibits the highest accuracy for budget parameters $B = 10$, 20 and 30.

As before, in Table 16 one can see that the proposed Approaches 1 and 3 have high features' overlap with the original MILP models for the LYMPHOMA dataset.

The results for LEUKEMIA dataset are summarized in Tables 17 and 18. From Table 17, one can observe that for the budget parameters $B = 20$ and 30, Approaches 1 and 3 perform faster compared to the original MILP models. For the case $B = 10$, Approach 1 performs faster than the original models. For the case $B = 50$, Approach 1 performs faster compared to the two original MILP models and Approach 3 exhibits a better running time only compared to the MILP2 model. According to this table, the best running times for the budget parameters B =10, 30 and 50 correspond to $MILP1^*$, while $MILP2^*$ has the shortest running time for $B = 20$. In terms of accuracy, $K\text{-}Inf\text{-}FS_s$ and $MILP2_s^*$ for budget parameter B =10 and 20 have the best accuracies, respectively. For the cases B =30 and 50 MILP2* has the best

Table 16: Features overlap percentage on LYMPHOMA dataset.

B	10	20	30	50
$F_{MILP1} \cap F_{MILP1^*}$	92*	86.7	86	80.5
$F_{MILP2} \cap F_{MILP2^*}$	90.7	**97.6**	**95.9**	**88.7**
$F_{MILP1^*} \cap F_{MILP2^*}$	90	88	87.8	81.1
$F_{MILP1^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	62.5	63.5	64.6	66.6
$F_{MILP2^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	64.6	66.7	64.6	63.2
$F_{MILP1_s^*} \cap F_{MILP1}$	86	91.6*	79.2	73.6
$F_{MILP1_s^*} \cap F_{MILP1}$	87.2	79.4	88.6	79.5
$F_{MILP1_s^*} \cap F_{MILP2_s^*}$	**100**	88.7	93.7*	82.6*
$F_{MILP1_s^*} \cap F_{Inf\text{-}FS_s}$	69.2	76.1	74.3	68.8
$F_{MILP2_s^*} \cap F_{Inf\text{-}FS_s}$	69.2	80.4	78.6	63.2

Table 17: Time and accuracy comparison for LEUKEMIA dataset.

Model		$B=10$	$B=20$	$B=30$	$B=50$
MILP1	Accuracy	91.1	92.6	93.6	92.6
	Time	97.6	10000	261	169.2*
MILP1*	Accuracy	90.6	93.3	97.4*	95.1
	Time	**48.8**	199	**88.6**	**44**
MILP2	Accuracy	90.9	91.6	97.1	95.3
	Time	163	10000	10000	10000
MILP2*	Accuracy	91.8	92.7	**97.6**	**97.2**
	Time	81.8*	**142**	10000	10000
K-Inf-FS$_s$	Accuracy	**95.8**	94.2	96.6	96.6*
	Time	195.3	188.9*	197.2*	196
MILP1$_s^*$	Accuracy	90.5	94.8*	96.8	93.2
	Time	192.3	216.2	200.6	205.9
MILP2$_s^*$	Accuracy	92.4*	**96.5**	97	95
	Time	192.6	222.6	1141	3527

accuracies.

According to the Table 18 for the LEUKEMIA dataset, one can see that the proposed Approaches 1 and 3 have high overlap with the original MILP models in most cases.

For the PROSTATE dataset the results are summarized in Tables 19 and 20. One can see that the proposed Approaches 1 and 3 perform faster than the original MILP models for the budget parameters B =30 and 50. For the first two budget parameters $B=10$ and 20, Approach 1 performs faster. It can also be seen from Table 19 that MILP2* has the best running time for all budget parameters. From the accuracy point of view, one can see that K-Inf-FS$_s$ has the best accuracy for the cases B =10 and 20, while for the budget parameters B =30 and 50 MILP1 and MILP2* have the best accuracies, respectively.

Finally, similar to the previous datasets, one can see that the proposed Approaches 1 and 3 have high features' overlap with the original MILP models as reported in Table 20 for the PROSTATE dataset.

Table 18: Features overlap percentage on LEUKEMIA dataset.

B	10	20	30	50
$F_{MILP1} \cap F_{MILP1^*}$	93.8*	79.9*	68.5	47.1
$F_{MILP2} \cap F_{MILP2^*}$	**100**	**91.3**	**77.5**	73.6
$F_{MILP1^*} \cap F_{MILP2^*}$	87.5	75.4	52.6	32.1
$F_{MILP1^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	66.7	67.1	52.2	55.9
$F_{MILP2^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	66.7	65.7	69.4	50
$F_{MILP1_s^*} \cap F_{MILP1}$	83.4	70.2	59.2	36.8
$F_{MILP2_s^*} \cap F_{MILP2}$	88.2	82.6	65.7	63.2
$F_{MILP1_s^*} \cap F_{MILP2_s^*}$	85.9	75.4	73.5*	66.2
$F_{MILP1_s^*} \cap F_{Inf\text{-}FS_s}$	72.7	67.1	67.5	**82.3**
$F_{MILP2_s^*} \cap F_{Inf\text{-}FS_s}$	72.7	65.7	63.9	78.2*

Table 19: Time and accuracy comparison for PROSTATE dataset.

Model		$B=10$	$B=20$	$B=30$	$B=50$
MILP1	Accuracy	85.7	85.6	**89.9**	92.6
	Time	104	134.5	204.8	431.8
MILP1*	Accuracy	85.6	86.8*	88.7	92.8*
	Time	69	74.5	97.1	121.6
MILP2	Accuracy	86*	85.8	89.4	92.6
	Time	44*	61.8*	126.3	417.6
MILP2*	Accuracy	85.9	86.2	89.6*	**92.9**
	Time	**21**	**22.3**	**24.9**	**46.2**
K-Inf-FS$_s$	Accuracy	**90.6**	**87.7**	86.2	87.5
	Time	93.7	85.5	95*	120.2*
MILP1$_s^*$	Accuracy	84.1	85.8	88.7	91.5
	Time	122.7	123.9	115.6	124.9
MILP2$_s^*$	Accuracy	84.2	85.3	89	92.3
	Time	111.1	111.9	113.4	122.5

5 Conclusions and future works

In this paper, first two MILP models for cost-effective feature selection and Inf-FS$_s$ model for feature ranking are reviewed. Then three hybrid approaches are proposed for cost effective feature selection. In the first approach, the relaxed linear version of the MILP model is solved first. Then, the MILP model for nonzero features weights ($|w_j| > 10^{-6}$) is solved, namely an MILP with less 0-1 variables. Then Inf-FS$_s$ is combined with a knapsack problem to choose cost effective feature. In the third approach, first Inf-FS$_s$ method is applied to rank the features. Then the first approach is applied to the original MILP models, restricted to the top 20% of features ranked by Inf-FS$_s$ method. The most important characteristic of the proposed approaches is that they are faster in terms of running time compared to the original MILP models, while they share most of the selected features.

As a continuation of this research, the following suggestions can be considered as potential future research directions:

Table 20: Features overlap percentage on PROSTATE dataset.

B	10	20	30	50
$F_{MILP1} \cap F_{MILP1^*}$	93.2*	93.3*	88.4	84.3*
$F_{MILP2} \cap F_{MILP2^*}$	**93.8**	**97.1**	89.5*	84.1
$F_{MILP1^*} \cap F_{MILP2^*}$	91.5	89.2	87	82.6
$F_{MILP1^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	33.3	63.5	60.5	56.1
$F_{MILP2^*} \cap F_{K\text{-}Inf\text{-}FS_s}$	33.3	63.5	58	56.8
$F_{MILP1_s^*} \cap F_{MILP1}$	92.4	91.2	83.6	79.2
$F_{MILP2_s^*} \cap F_{MILP2}$	90.2	94.6*	84.4	78.2
$F_{MILP1_s^*} \cap F_{MILP2_s^*}$	89.8	84	**90.8**	**85.8**
$F_{MILP1_s^*} \cap F_{Inf\text{-}FS_s}$	61.9	75.9	77.3	60.8
$F_{MILP2_s^*} \cap F_{Inf\text{-}FS_s}$	61.9	75.9	78.8	62.7

- Incorporating budget constraints into the structure of the Inf-FS$_s$ method.
- Investigating and evaluating the robustness of the proposed approaches against data uncertainty.
- Exploring and testing the proposed approaches for nonlinear separation problems.
- Combining the proposed approaches with the clustering Inf-FS$_s$ method proposed in [10].

References

[1] U. Alon, N. Barkai, D.A. Notterman, A.J. Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, Proc. Natl. Acad. Sci. U.S.A **96** (1999) 6745–6750.

[2] G.C. Berens, R.L. Houghton, D.C. Seitz, E.R. Ruiz, *Gene expression correlates of clinical prostate cancer behavior*, Cancer Cell **1** (2002) 203–209.

[3] P. Bradley, O. Mangasarian, *Feature selection via concave minimization and support vector machines*, Proceedings of the Fifteenth International Conference on Machine Learning (ICML), Morgan Kaufmann (1998) 82–90.

[4] G. Chandrashekar, F. Sahin, *A survey on feature selection methods*, Comput. Electr. Eng., **40** (2014) 16–28.

[5] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, Second Edition, Wiley, 2000.

[6] M. Grant, S. Boyd, CVX: MATLAB software for disciplined convex programming, Version 2.1 , March 2017.

[7] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C.D. Bloomfield, E.S. Lander, *Molecular classification of cancer, Class discovery and class prediction by gene expression monitoring*, Science **286** (1999) 531–537.

[8] I. Guyon, A. Elisseeff, *An introduction to variable and feature selection*, J. Mach. Learn. Res. **3** (2003) 1157–1182.

[9] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, *Feature Extraction, Foundations and Applications*, Springer, 2006.

[10] S.F. Hassani Ziabari, S. Eskandari, M. Salahi, *CInf-FS$_S$: An efficient infinite feature selection method using K-means clustering to partition large feature spaces*, Pattern Anal. Applic. **26** (2023) 1631–1639.

[11] M. Labb, L.I. Martnez-Merino, A.M. Rodrguez-Cha, *Mixed integer linear programming for feature selection in support vector machine*, Discrete Appl. Math. **319** (2018) 1–7.

[12] I.G. Lee, Q. Zhang, S.W. Yoon, D. Won, *Mixed integer linear programming support vector machine for cost-effective feature selection*, Knowl-Based Syst. **193** (2020) 1–6.

[13] S. Maldonado, J. Prez, R. Weber, M. Labb, *Feature selection for support vector machines via mixed integer linear programming*, Inf. Sci. **279** (2014) 163–175.

[14] S. Maldonado, R. Weber, J. Basak, *Kernel-penalized SVM for feature selection*, Inf. Sci **181** (2011) 115–128.

[15] G. Roffo, S. Melzi, U. Castellani, A. Vinciarelli, M. Cristani, *Infinite feature selection: A graph-based feature filtering approach*, IEEE Trans. Pattern Anal. Mach. Intell. **43** (2021) 4396–4410.

[16] W. Zhou, L. Zhang, L. Jiao, *Linear programming support vector machines*, Pattern Recognit. **35** (2002) 2927–2936.