Discriminant analysis and feature selection for emergency department readmission prediction

Daniëlle Hooijenga[1], Raksmey Phan[1], Vincent Augusto[1], Xiaolan Xie[1], and Abdesslam Redjaline[2]

[1]Mines Saint-Etienne, Univ Clermont Auvergne CNRS, UMR 6158 LIMOS, F - 42023 Saint-Etienne France
[danielle.hooijenga;raksmey.phan;augusto;xie]@emse.fr
[2]Hôpital Le Corbusier, Firminy, France

***Abstract*—Readmission to the emergency department is an important indicator of the quality of care. By estimating the risk of readmission, measures may be taken and revisits avoided. In this study we use an optimization-based discriminant analysis model (DAMIP), in combination with different heuristics for feature selection, to predict emergency department readmission within 72 hours. For feature selection random search, particle swarm optimization, and tabu search are considered. The model offers the possibility to not classify some patients in order to ensure acceptable prediction accuracy. A balancing technique is used to generate balanced training sets to improve the prediction of the small readmission group.**

The proposed method is tested on French healthcare data including diagnosis-related group (DRG) and patient characteristics of over 12 million emergency stays, for which approximately 5% of the patients return within 72 hours to the emergency department. We compare the results to different classification algorithms, including linear discriminant analysis, naive Bayesian, support vector machine, logistic regression, classification tree, random forest, nearest shrunken centroid, and neural network. The proposed method is shown to achieve similar or higher prediction accuracies than classical classification algorithms.

***Index Terms*—machine learning, DAMIP, classification, feature selection, hospital readmission**

I. Introduction

In this article we consider a classification problem, with the goal of predicting whether a patient will visit the emergency department (ED) again after being discharged from the ED earlier. Readmission to the emergency department is often considered a measure of quality in healthcare [3]. A considerable part of readmissions have been judged to be preventable. By estimating the risk of readmission, measures may be taken and revisits avoided.

The studied classification problem consists of using patient characteristics, so-called features, to make a prediction on the probability of readmission for new incoming patients at the emergency department. Possible features could be age, gender, and diagnosis. In healthcare data, many characteristics are available, but not necessarily all of them are relevant for prediction. This makes feature selection relevant to reduce the complexity and improve the results. One of the challenges in the prediction of emergency department readmission is the fact that the data is highly imbalanced. More than 95% of the patients do not return to the emergency department after their visit.

In this article, we apply an optimization approach of supervised classification to French healthcare data in combination with three different feature selection algorithms: random search, particle swarm optimization, and tabu search. Specifically, our contribution lies in the development of the tabu search approach for feature selection including a sampling strategy to speed up the learning process. To validate the relative performance of the applied techniques, the results are compared to classical classification algorithms in machine learning.

The remainder of this article is structured as follows. In Section II we introduce relevant literature. The necessary notation and an explanation of the used methods are given in Sections III and IV. An overview of the complete framework as developed is given in Section V. In Section VI we describe the numerical results and Section VII gives a conclusion and discussion on the studied topic.

II. Literature Review

The Discriminant Analysis Mixed Integer Programming (DAMIP) model was first introduced by [11]. The authors apply Particle Swarm Optimization (PSO) for feature selection and combine this with the DAMIP model for classification. Their method is applied to US healthcare data on emergency department readmission and the authors show that their method outperforms other classification methods.

Emergency department readmission in the United States has also been studied by [17]. The author gives a linear optimization model for classification, as well as several variations of this model. The presented framework is applied to emergency department readmission, but also to data on flu vaccine responders, knee reinjections, and Alzheimer's disease.

Classification problems have been studied extensively. [10] give an overview of several supervised machine learning algorithms for classification problems. The authors provide an explanation of algorithms like decision trees, support vector machines, and Bayesian networks. The advantages and disadvantages of each algorithm are discussed, but, as the authors state, the choice of an algorithm always depends on the task considered.

A widely studied subject in the area of classification is the topic of feature selection. [8] provide a review of algorithms

978-1-5386-9276-9/18/$31.00 ©2018 IEEE

used to select features in the application on microarray data. The authors describe different methods of feature selection, like genetic algorithms and random forests, and discuss their advantages and disadvantages.

Particle swarm optimization is applied for feature selection by [6]. As a fitness value, the authors use the accuracy of a support vector machine classifier. It is shown that this approach can select the most informative features, based on classification accuracy within a reasonable CPU time.

[18] apply tabu search to a feature selection problem. The authors compare the results to classical feature selection methods such as sequential forward selection and sequential backward selection. It is shown that tabu search often finds the optimal solution, where other methods suffer from being trapped into local optimal solutions.

Tabu search is also applied for feature selection by [14]. The authors propose a tabu search procedure with a long-term memory. Besides the long-term memory, the proposed algorithm also considers a broader neighbourhood than the tabu search algorithm used by, for example, [18].

In this article, we propose a new supervised classification approach based on DAMIP taking into account specific health-care data and a new heuristic for feature selection based on tabu search.

III. Classification model

In this section, we show the optimization-based model for discriminant analysis (DAMIP) as was presented in [11]. The relevant notation is given in Table I.

TABLE I
Notation for DAMIP

Symbol	Description
R_k	Region assigned to group k
R_0	Region for "deferred judgment"
π_k	Prior probability of group k
$f_k(x)$	Conditional probability function of group k
α_{hk}	Upper bound on misclassification where the observations of group k are classified to group h
λ_{hk}	Non-negative constants giving the optimal decision rule
u_{ki}	Decision variable; equals one if entity i is classified to group k and zero otherwise
y_i	Group to which entity i belongs
L_{ki}	Loss functions

Anderson [1] proposes to seek for a partition $\{R_0, R_1, ..., R_K\}$, where R_k is the region assigned to group k and R_0 is a region for "deferred judgment". This region is introduced to be able to put a restriction on the probability of misclassification. The model proposed by Anderson to find the described partition is as follows.

$$\text{Max} \quad \sum_{k \in \mathcal{K}} \pi_k \int_{R_k} f_k(x)dx \tag{1}$$

$$\text{s.t.} \quad \int_{R_h} f_k(x)dx \leq \alpha_{hk} \qquad \forall h,k \in \|, h \neq k \tag{2}$$

where π_k is the prior probability of group k, $f_k(x)$ is the conditional probability density function of group k and α_{hk} is the predetermined limit on the misclassifications where the observations of group k are classified to group h.

Anderson showed that there exist non-negative constants $\lambda_{hk}, h, k \in K, h \neq k$, such that the optimal decision rule is given by

$$R_k = \{x \in \mathbb{R}^m : L_k(x) = \max_{h \in \{0\} \cup \mathcal{K}} (L_h(x))\}, k \in \{0\} \cup \mathcal{K} \tag{3}$$

where

$$L_0(x) = 0 \tag{4}$$

$$L_k(x) = \pi_k f_k(x) - \sum_{h \in \mathcal{K}, h \neq k} \lambda_{hk} f_h(x), k \in \mathcal{K} \tag{5}$$

Below the formulation of the Discriminant Analysis Mixed Integer Programming (DAMIP) model, as presented by [11], is given.

$$\text{Max} \quad \sum_{i \in \mathcal{O}} u_{y_i i} \tag{6}$$

$$\text{s.t} \quad L_{ki} = \pi_k f_k(\mathbf{x}_i) - \sum_{h \in \mathcal{G}, h \neq k} f_h(\mathbf{x}_i)\lambda_{hk} \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{7}$$

$$u_{ki} = \begin{cases} 1 & \text{if } k = argmax\{0, L_{hi} : h \in \mathcal{G}\} \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{O}, \forall k \in \{0\} \cup \mathcal{G} \tag{8}$$

$$\sum_{k \in \{0\} \cup \mathcal{G}} u_{ki} = 1 \quad \forall i \in \mathcal{O} \tag{9}$$

$$\sum_{i : i \in \mathcal{O}_h} u_{ki} \leq \lfloor \alpha_{hk} n_h \rfloor \quad \forall h, k \in \mathcal{G}, h \neq k \tag{10}$$

$$u_{ki} \in \{0, 1\} \quad \forall i \in \mathcal{O}, \forall k \in \{0\} \cup \mathcal{G} \tag{11}$$

$$L_{ki} \in \mathbb{R} \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{12}$$

$$\lambda_{hk} \geq 0 \quad \forall h, k \in \mathcal{G}, h \neq k \tag{13}$$

The decision variable u_{ki} equals one if entity i is classified to group k and zero otherwise. The parameter y_i gives the group to which entity i truly belongs. That is, $u_{y_i i}$ equals to 1 if entity i is classified correctly. The objective (6) of the model is to maximize the number of correctly classified entities. π_k is the prior probability of an entity belonging to group k. $f_k(\mathbf{x}_i)$ represents the conditional probability density function. It is defined to be the probability of the features having the values given in $\mathbf{x}_i$, given that the entity is in group k. λ_{hk} are the non-negative constants such that the optimal decision rule is determined, this is a decision variable. Constraints (7) and (8) determine the classification of the entities, the entity is classified to the group k, for which L_{ki} is maximum, and to the unclassified group if the maximum is negative. Constraint (9) makes sure that every entity is assigned to exactly one group. Constraint (10) puts an upper bound on the allowed rate of misclassification, where α_{hk} is a parameter which is to be set by the user, and n_h is the number of entities in group h.

The non-linearity of the model makes it more impractical to solve. [17] proposes a linear version of the model, which is given below.

$$\text{Max} \quad \sum_{i \in \mathcal{O}} u_{y_i i} \tag{14}$$

$$\text{s.t} \quad L_{ki} = \pi_k f_k(\mathbf{x}_i) - \sum_{h \in \mathcal{G}, h \neq k} f_h(\mathbf{x}_i) \lambda_{hk} \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{15}$$

$$a_i - L_{ki} \leq M(1 - u_{ki}) \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{16}$$

$$a_i \leq M(1 - u_{0i}) + \epsilon \quad \forall i \in \mathcal{O} \tag{17}$$

$$a_i - L_{ki} \geq \epsilon(1 - u_{ki}) \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{18}$$

$$a_i \geq \epsilon u_{ki} \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{19}$$

$$\sum_{k \in \{0\} \cup \mathcal{G}} u_{ki} = 1 \quad \forall i \in \mathcal{O} \tag{20}$$

$$\sum_{i : i \in \mathcal{O}_h} u_{ki} \leq \lfloor \alpha_{hk} n_h \rfloor \quad \forall h, k \in \mathcal{G}, h \neq k \tag{21}$$

$$u_{ki} \in \{0, 1\} \quad \forall i \in \mathcal{O}, \forall k \in \{0\} \cup \mathcal{G} \tag{22}$$

$$L_{ki} \in \mathbb{R} \quad \forall i \in \mathcal{O}, \forall k \in \mathcal{G} \tag{23}$$

$$\lambda_{hk} \geq 0 \quad \forall h, k \in \mathcal{G}, h \neq k \tag{24}$$

$$a_i \geq 0 \quad \forall i \in \mathcal{O} \tag{25}$$

In this model, ϵ and M represent a small and a large number, respectively. The constraint set (8) of the non-linear model is replaced by constraint sets (16) - (19) in the linear model. These constraints make sure that the considered entity is classified into the group with the highest value of L_{ki} or in the group of reserved judgment if all values of L_{ki} are negative.

IV. Feature selection

In this section, the three different proposed heuristics for feature selection are presented.

As not all features might be relevant to the prediction, we need a way to determine which set of features will give the best prediction. Three types of feature selection algorithms can be distinguished: filter algorithms, wrapper algorithms, and embedded algorithms. In filter algorithms, features are extracted from the data without any involvement of learning. In wrapper approaches, learning techniques are used to evaluate which features are useful. The embedded algorithms combine the process of feature selection with the construction of the classifier [8]. In this article we focus on wrapper algorithms for feature selection, where sets of features are evaluated by the performance of the classifier.

A. Random search

Random search is a search algorithm in which random solutions are created for a specified number of iterations. The best found solution among these is given as the output of the algorithm. Below an overview of the algorithm is given.

For k iterations:

- Draw random number n in [1,*nrFeatures*], where *nrFeatures* is the initial number of features
- Select n features at random
- Run DAMIP with selected set of features
- If objective value is better than the best so far, best so far becomes current solution

B. Particle swarm optimization

Particle Swarm Optimization (PSO) is an evolutionary computation technique for solving optimization problems, first introduced by Kennedy and Eberhart [9]. PSO optimizes a problem by iteratively trying to improve a candidate solution, represented by particles. Each particle's movement is influenced by its best known solution and by the best known solution of its neighbours. The relevant notation, specific for particle swarm optimization, is given below, in Table II.

TABLE II
Notation for PSO

Symbol	Description
$\mathbf{x}_i$	Vector of features for particle i
$\mathbf{v}_i$	Velocity of particle i
a_i	Objective value of best found solution of particle i
a'	Objective value of current solution
$\mathbf{p}_i$	Features of best found solution of particle i
$N(i)$	Neighbourhood of particle i
$p_{N(i)}$	Features of the best solution in the neighbourhood of particle i

Below, an overview of the PSO/DAMIP framework is given. $\mathbf{x}_i$ represents the features which are used in an iteration for particle i, it is a vector of length m, with m the total number of features. $\mathbf{v}_i$ represents the velocity of particle i and has the same size as $\mathbf{x}_i$. The best achieved objective value so far is given by a_i and the objective value in the current iteration is given by a'. The features belonging to the best achieved objective are given in $\mathbf{p}_i$. Furthermore, $N(i)$ represents the neighbourhood of particle i, and $\mathbf{p}_{N(i)}$ gives the features belonging to the best achieved objective value so far in the neighbourhood of particle i. The particles are represented by a grid. The neighbourhood is defined by the particles to the top, bottom, left, and right of the concerned particle.

1) Initialization
   For every particle i:
   - $\mathbf{x}_i$ is randomly generated by selecting k 1's at random, where k is the desired number of features
   - $\mathbf{v}_i$ is randomly generated with numbers between $-V_{max}$ and V_{max}, with V_{max} predetermined
   - $\mathbf{p}_i \leftarrow \mathbf{x}_i$
   - $a_i \leftarrow 0$

2) Update
   For every particle i:
   - Run DAMIP model with features $\mathbf{x}_i$
   - If the objective value $a' > a_i$:
     - $\mathbf{p}_i \leftarrow \mathbf{x}_i$
     - $a_i \leftarrow a'$
   - $\mathbf{v}_i \leftarrow \omega\mathbf{v}_i + c_1r_1(\mathbf{p}_i - \mathbf{x}_i) + c_2r_2(\mathbf{p}_{N(i)} - \mathbf{x}_i)$, where ω, c_1, c_2 are fixed positive coefficients and r_1 and r_2 are randomly generated in the range $(0, 1)$
   - $\mathbf{x}_i$ is determined by selecting the k features with the largest values in $\mathbf{v}_i$
3) Termination
   - Maximum number of iterations is reached

C. Tabu search

Tabu search, first introduced in [7], is a local search algorithm, which tries to escape possible local minima by keeping track of a list of forbidden solutions (the tabu list). The specific notation for tabu search is given below, in Table III.

TABLE III
NOTATION FOR TABU SEARCH

Symbol	Description
$\mathbf{x}$	Initial solution
c	Size of candidate set
$C(\mathbf{x})$	Candidate set of solution $\mathbf{x}$
$\mathbf{y}$	Best solution in $C(\mathbf{x})$
l	Tabu list length
a_i	Objective value of best found solution
a'	Objective value of current solution
$\mathbf{p}_i$	Features of best found solution

In the algorithm, we start with an initial solution $\mathbf{x}$, which represents a specific set of features. For this initial solution, the criterion function is evaluated. Next, a set of candidate moves is considered. A candidate move is the move from one subset of features to another subset of features, where exactly one feature differs in presence or absence. We take into consideration c candidate moves, where c is a parameter specified by the user. If the best of these moves is not in the tabu list, this solution is now considered to be the current solution and this solution is placed in the tabu list (TL), which has length l. The tabu list prevents moves to be reversed within l iterations, l to be set by the user. The procedure is repeated for a specified number of iterations. Similar to before, $\mathbf{p}$ represents the features used in the best found solution so far and a is the corresponding objective value.

Algorithm

1) Initialization
   - Generate an initial solution $\mathbf{x}$
   - TL $\leftarrow \emptyset$
   - $\mathbf{p} \leftarrow \mathbf{x}$
   - a $\leftarrow 0$
2) Update
   - Randomly pick c solutions from the neighbourhood of $\mathbf{x}$ to form candidate set $C(\mathbf{x})$
   - If $C(\mathbf{x}) = \emptyset$, regenerate $C(\mathbf{x})$
     Otherwise, find best solution $\mathbf{y}$ in $C(\mathbf{x})$
   - If $\mathbf{y} \in TL$, $C(\mathbf{x}) \leftarrow C(\mathbf{x}) - \mathbf{y}$ and find next best solution
     Otherwise, $\mathbf{x} \leftarrow \mathbf{y}$. If $a' > a$: $a \leftarrow a'$ and $\mathbf{p} \leftarrow \mathbf{y}$
   - $TL \leftarrow TL \cup \mathbf{x}$, if the length of TL exceeds l, remove the head item of the list
3) Termination
   - Maximum number of iterations is reached

V. Classification Framework

In Figure 1 an overview of the framework of DAMIP in combination with feature selection is given. Starting from the complete data set, we use a balancing technique to arrive at a balanced data set. Note that in this data set only the training data is balanced, for a representative evaluation the testing data remains imbalanced. From the balanced data set, we take samples of data to use throughout the feature selection heuristics. DAMIP is applied to the samples of data, with different subsets of features for the purpose of feature selection. Note that applying DAMIP involves two parts: training and testing. The results from the testing part are the results used to be able to compare solutions to each other. For the best performing subset of features, DAMIP is run with the complete balanced data set. This process is repeated 100 times and at the end of the procedure the output is the best found set of features, with the corresponding accuracies of classification.

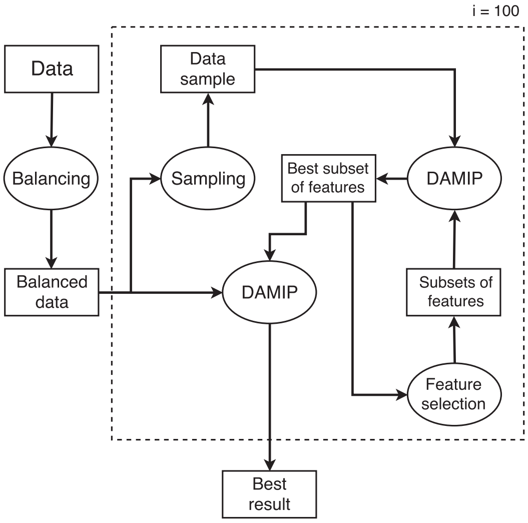

Fig. 1. An overview of the framework

1) Balancing the data: As was mentioned before, the data which is used is highly imbalanced. The majority of the entities belongs to the not-return group. In order to train the model well, we create a more balanced data set by means of under-sampling. Concretely, from the data set, we delete a part of the entities from the majority class, the not-return class. In the balanced data set, there are an equal amount of return entities as non-return entities. Note that we only use the balanced data set for the training part of the DAMIP model, for fair evaluation the testing part is performed on the imbalanced data set.

2) Sampling: To restrict the computational time, we make use of sampling for tabu search and for particle swarm optimization. In this approach, we make use of a sample from the data set to get an indication of performance. For tabu search, we run DAMIP on the complete data set once every iteration. For particle swarm optimization DAMIP is applied to all data every time a new best solution is found. For one sample data set, we select at random 10% of the complete data set.

VI. Results

In this section we give an overview of the used data set for the numerical results, followed by the benchmark setup consisting of several classical classification algorithms. Finally, we provide the numerical results both on the classical algorithms, as well as on the developed framework.

A. Data

The data used in this study consists of 91 000 emergency department admissions, which represents approximately one month of emergency department admissions. Among these admissions, around 5% of the patients return within 72 hours. Features known for each of the admissions include the age of the patient, the gender, the arrival mode, the urgency level, the length of stay, and the main diagnosis. In total 44 features are taken into account, of which 22 features represent the main diagnosis. These 22 features are binary and indicate if the diagnosis in a certain category was given. Some of the features are not binarized as this is not a prerequisite for DAMIP. An example of such feature is the urgency level, which ranks from 1 to 5.

Among the 91 000 admissions, 54.7% are men. The vast majority (97.1%) of the patients arrived from home (in contrast to arriving from a medical unit). Moreover, after treatment at the emergency department, most patients (78.1%) return home. The most common main diagnosis is a traumatic injury (44.6%).

B. Benchmark setup

In this section we provide a benchmark to compare our developed framework to classical classification methods listed in Table IV. Algorithms are shortly explained below.

Linear discriminant analysis is a commonly used technique for data classification. The method tries to maximize the ration of between-class variance to the within-class variance, guaranteeing maximal separability [2]. A Naive Bayesian classifier based on Bayes' theorem is a probabilistic statistical classifier [16]. This classifier is based on the assumption that all features are independent of each other. Fundamentally, support vector machines search for the optimal separating hyperplane, where the margin between two different objects is maximal. To find this maximal margin, support vectors are used [16]. Logistic regression is a statistical regression model, which has as an advantage that it provides the user explicitly with probabilities and not only the class label information [13]. Classification tree classifiers construct a tree structure, where at every step an attribute is sought whose sorting result is closest to the pure partitions by the class in terms of class values [16]. Random forest was introduced in [4]. This algorithm uses a group of classification tress, each of which is built using a bootstrap sample of the data [5]. In the nearest shrunken centroid algorithm for classification, shrunken centroids are used for each class and test samples are classified to the class whose shrunken centroid is nearest to it [15]. Neural network attempts to mimic the neurological functions of the brain [16]. Neural network consists of nodes mimicking the functions of neurons in the brain. The nodes are interconnected via links with adjustable weights. The weights are adjusted by learning.

C. Numerical results

In this section, the results of both DAMIP in combination with different feature selection algorithms as well as other classification algorithms are presented. The model is trained using 80% of the data, the model is thereafter tested on the remaining 20% for all algorithms. All the described algorithms are implemented in the Python language, where the classical classification algorithms have been implemented with help of the scikit-learn package [12], using the parameters given as default by the package.

1) Classical classification algorithms: In Table IV the results of the different classical classification algorithms are shown. The following metrics are considered: (i) *Accuracy* represents the overall accuracy of the solution (percentage of correctly predicted entities over all the classified entities); (ii) *Specificity* is the percentage of correctly predicted non-return (true negative rate); (iii) *Recall* (sensitivity) is the percentage of correctly predicted return (true positive rate); (iv) *Precision* is the positive predictive value; and (v) *F1-score* is an overall quality measure combining precision and recall, defined as follows:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{26}$$

Almost all of the shown classical supervised classification methods tend to classify all entities into the non-return group, as this will give a high overall accuracy. This also shows the importance of providing the accuracy of the different classes (sensitivity and specificity) as the overall accuracy may give a distorted image of performance. The only two algorithms succeeding in classifying some entities to the return group are naive Bayesian and nearest shrunken centroid, which achieve the best F1-score (respectively 0.148 and 0.117). However, for naive Bayesian, recall remains very low (28.7%). Nearest shrunken centroid shows better results, but with a high impact on the specificity (48.4%). In summary, none of the considered methods are suitable for use by health practitioners as a

TABLE IV
RESULTS OF CLASSICAL CLASSIFICATION ALGORITHMS

Method	Accuracy	Specificity (non-return)	Recall (return)	Precision	F1-score
Linear Discriminant Analysis	95.3%	100.0%	0.0%	*NA*	*NA*
Naive Bayesian	83.7%	86.4%	28.7%	9.9%	0.148
Support Vector Machine	95.3%	100.0%	0.0%	*NA*	*NA*
Logistic Regression	95.3%	100.0%	0.0%	*NA*	*NA*
Classification Tree	95.0%	99.7%	0.0%	*NA*	*NA*
Random Forest	95.1%	99.7%	0.1%	15.1%	0.018
Nearest Shrunken Centroid	49.3%	48.4%	67.5%	6.4%	0.117
Neural Network	95.3%	100.0%	0.0%	*NA*	*NA*
DAMIP	72.0%	74.1%	34.4%	6.5%	0.110

reliable decision aid tool for detecting patients having a high probability of returning to the emergency department.

2) Feature selection: In Table V the results of DAMIP in combination with the different feature selection algorithms are given. The first two lines, show the results of only DAMIP, using all given features. The other lines give the results of DAMIP with the shown feature selection algorithm. For each of the algorithms, 100 iterations were run. The parameters which need to be set by the user have been chosen by numerical experimentation. In the feature selection heuristics, solutions are compared to each other by the value of recall. That is, the accuracy of the return group. This is done because if we compare by overall accuracy, it will be more beneficial for the heuristics to choose a solution with all entities classified to the non-return group, as this results in high overall accuracy.

For each of the methods, two lines of results are presented. One line where all entities have been classified and one line where part of the entities may not be classified. By not classifying all entities we can achieve a better performance, by leaving the uncertain entities unclassified. In the column *#Features*, the number of selected features are given. Clearly, for the DAMIP model without feature selection, all the features are used.

From the table we can see that we can achieve the best results by using tabu search for feature selection (best F1-score with 0.157), followed shortly by particle swarm optimization (F1-score 0.147), especially by leaving a part of the entities unclassified. When forbidding non-classification, DAMIP combined with tabu search achieves a precision greater than 50% for both groups (F1-score 0.137). When allowing non-classification, DAMIP combined with particle swarm optimization has less unclassified entities, but a lower overall accuracy than our approach. Both algorithms show a better performance in terms of classifying to the return group, as compared to not applying feature selection.

Concerning the number of selected features by the different heuristics, tabu search shows the highest number of selected features. The feature which is selected most frequent is the *mode of exit* of the patient, which means if the patient goes home or to another hospital department. Contrary to the mode of exit, the *mode of arrival* is never selected by the feature selection heuristics. All other features are selected at most by half of the six feature selection heuristics.

3) Sampling: As mentioned before, we make use of a sampling technique in order to reduce the complete computational time. In the feature selection algorithms, most computational time is due to the evaluation of the different subsets of features performed by the DAMIP model. By using a sample of the data we can largely decrease the computational time. To get some insight in the decrease in computation time, we performed one iteration of DAMIP, including training and testing, with the full data set and one iteration with a sampled data set. The results are given in Table VI. We can see that the quality of the results remain equivalent. However, the computational time, which is shown in the last column, has decreased significantly, from more than one minute to less than one second.

TABLE VI
RESULTS OF DAMIP WITHOUT AND WITH SAMPLING

Sampling	Accuracy	Specificity	Recall	CPU (s)
No	73.6%	76.0%	28.0%	60.49
Yes	79.1%	81.6%	25.0%	0.71

VII. DISCUSSION AND CONCLUSION

In this article we have considered the problem of predicting emergency department readmission. We applied DAMIP as a classifier to the data set and considered multiple heuristics for feature selection. The results of the proposed method were compared to those of classical machine learning algorithms and it was shown that the developed DAMIP framework based on the tabu search algorithm for feature selection can beat the performance of these standard algorithms. Moreover, we have used a sampling technique, which shows a big decrease in computational time while achieving a similar level of results.

Although our method outperforms existing classification methods, precision and F1-scores remain very low. The unbalanced nature of the dataset explains such low results. In future research it may be interesting to look deeper into the possibilities of feature selection. The performance of tabu search for feature selection may be improved by, for example, selecting the features in each iteration based on knowledge of the previous iterations, rather than selecting a neighbour at random.

Furthermore it would be interesting to test the proposed framework on a data set with more entities (real regional data set of emergency records includes more than 12 million

TABLE V
RESULTS OF DAMIP AND FEATURE SELECTION HEURISTICS

Feature selection method	**Accuracy**	**Specificity (non-return)**	**Recall (return)**	**Precision**	**Unclassified**	**#Features**	**F1-score**
None (DAMIP only)	72.0%	74.1%	34.4%	6.5%	0.0%	44	0.110
None (DAMIP only), with non-classification	67.6%	69.0%	42.7%	6.7%	6.8%	44	0.117
Random search	47.5%	46.9%	59.1%	5.5%	0.0%	7	0.101
Random search, with non-classification	45.5 %	44.5%	62.9%	5.6%	4.2%	4	0.103
PSO	43.1%	41.8%	66.7%	5.7%	0.0%	5	0.105
PSO, with non-classification	60.0%	59.5%	69.1%	8.2%	18.7%	15	0.147
Tabu search	67.6%	68.4%	51.6%	7.9%	0.0%	13	0.137
Tabu search, with non-classification	63.0%	62.7%	69.0%	8.9%	37.7%	19	**0.157**

emergency department admissions) and more features (by taking into consideration additional patient characteristics). Other balancing techniques should be considered in order to improve the precision of our method.

When the performance of the method is improved and more reliable predictions can be made, this will provide the staff of the emergency department with a reliable tool to foresee if a specific needs more extensive treatment. In this way readmission might be avoided, helping both increase the performance measures of the hospital as well as the well-being of the patient.

REFERENCES

[1] J.A. Anderson. Constrained discrimination between k populations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 123–139, 1969.

[2] S. Balakrishnama and A. Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18:1–8, 1998.

[3] J. Benbassat and M. Taragin. Hospital readmissions as a measure of quality of health care: advantages and limitations. *Archives of internal medicine*, 160(8):1074–1081, 2000.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] R. Díaz-Uriarte and S.A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[6] P. Ghamisi and J.A. Benediktsson. Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geoscience and Remote Sensing Letters*, 12(2):309–313, 2015.

[7] F. Glover. Tabu searchpart i. *ORSA Journal on computing*, 1(3):190–206, 1989.

[8] Z.M. Hira and D.F. Gillies. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015.

[9] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[10] S.B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[11] E.K. Lee, F. Yuan, D.A. Hirsh, M.D. Mallory, and H.K. Simon. A clinical decision tool for predicting patient care characteristics: patients returning within 72 hours in the emergency department. In *AMIA Annual Symposium Proceedings*, volume 2012, page 495. American Medical Informatics Association, 2012.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] S.K. Shevade and S.S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

[14] M.A. Tahir, A. Bouridane, and F. Kurugollu. Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier. *Pattern Recognition Letters*, 28(4):438–446, 2007.

[15] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, pages 104–117, 2003.

[16] I. Yoo, P. Alafaireet, M. Marinov, K. Pena-Hernandez, R. Gopidi, J.F. Chang, and L. Hua. Data mining in healthcare and biomedicine: a survey of the literature. *Journal of medical systems*, 36(4):2431–2448, 2012.

[17] F. Yuan. *Modeling and computational strategies for medical decision making*. PhD thesis, Georgia Institute of Technology, 2015.

[18] H. Zhang and G. Sun. Feature selection using tabu search method. *Pattern recognition*, 35(3):701–711, 2002.