symmetry

MDPI

Article

Formation of Fuzzy Patterns in Logical Analysis of Data Using a Multi-Criteria Genetic Algorithm

Igor S. Masich [1,2], Margarita A. Kulachenko [1], Predrag S. Stanimirović [3], Aleksey M. Popov [1], Elena M. Tovbis [1], Alena A. Stupina [1,4] and Lev A. Kazakovtsev [1,4,*]

[1] Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology, 31 Krasnoyarsky Rabochy av., 660037 Krasnoyarsk, Russia; i-masich@yandex.ru (I.S.M.); margaritakulachenko@yandex.ru (M.A.K.); vm_popov@sibsau.ru (A.M.P.); sibstu2006@rambler.ru (E.M.T.); h677hm@gmail.com (A.A.S.)
[2] Institute of Space and Information Technologies, Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia
[3] Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia; pecko@pmf.ni.ac.rs
[4] Institute of Business Process Management, Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia
* Correspondence: levk@bk.ru

Abstract: The formation of patterns is one of the main stages in logical data analysis. Fuzzy approaches to pattern generation in logical analysis of data allow the pattern to cover not only objects of the target class, but also a certain proportion of objects of the opposite class. In this case, pattern search is an optimization problem with the maximum coverage of the target class as an objective function, and some allowed coverage of the opposite class as a constraint. We propose a more flexible and symmetric optimization model which does not impose a strict restriction on the pattern coverage of the opposite class observations. Instead, our model converts such a restriction (purity restriction) into an additional criterion. Both, coverage of the target class and the opposite class are two objective functions of the optimization problem. The search for a balance of these criteria is the essence of the proposed optimization method. We propose a modified evolutionary algorithm based on the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) to solve this problem. The new algorithm uses pattern formation as an approximation of the Pareto set and considers the solution's representation in logical analysis of data and the informativeness of patterns. We have tested our approach on two applied medical problems of classification under conditions of sample asymmetry: one class significantly dominated the other. The classification results were comparable and, in some cases, better than the results of commonly used machine learning algorithms in terms of accuracy, without losing the interpretability.

Keywords: logical analysis of data; pattern generation; genetic algorithm

Citation: Masich, I.S.; Kulachenko, M.A.; Stanimirović, P.S.; Popov, A.M.; Tovbis, E.M.; Stupina, A.A.; Kazakovtsev, L.A. Formation of Fuzzy Patterns in Logical Analysis of Data Using a Multi-Criteria Genetic Algorithm. *Symmetry* **2022**, *14*, 600. https://doi.org/10.3390/sym14030600

Academic Editor: László T. Kóczy

Received: 25 February 2022
Accepted: 15 March 2022
Published: 17 March 2022

1. Introduction

Logical analysis of data (LAD) is a methodology for processing a set of observations, or objects, some of which belong to a specific subset (positive observations), and the rest does not belong to it (negative observations) [1]. These observations are described by features, generally numerical, nominal, or binary. Logical analysis of data is performed by detecting pattern-logical expressions that are true for positive (or negative) observations and not performed for negative (or, respectively, positive) observations [2]. Thus, regions in the feature space containing observations of the corresponding classes can be approximated using a set of positive and negative patterns. To identify such patterns, we use models and methods of combinatorial optimization [3–5].

Classification problems are one of the application fields of LAD [6,7]. From the point of view of solving classification problems, applying LAD can be considered as the construction of a rule-based classifier. Like other rule-based classification approaches, such as decision trees and lists of rules, this approach has the advantage that it constructs a "transparent" classifier. Thus, it belongs to interpretive machine learning methods.

Two types of patterns can be distinguished [8]. The first type is homogeneous (pure, clear) patterns. A homogeneous pattern covers part of the observations of a particular class (for example, positive) and does not cover any observation of another class (negative). However, if we consider real data for constructing a classifier, then pure patterns often do not give a good result. Due to the noisiness of the data, the presence of inaccuracies, errors, and outliers, pure patterns may have too low generalizing ability, and their use leads to overfitting. In such cases, the best results are shown by fuzzy (partial) patterns, in which the homogeneity constraint is weakened. Such weakening (relaxation) leads to the formation of more generalized patterns [4,9]. The pattern search problem is considered as an optimization problem, where the objective function is the number of covered observations of a certain class under a relaxed non-coverage constraint of observations of the opposite class.

Modern literature offers various approaches to the formation of patterns [8]. To generate patterns, enumeration-based algorithms [6,10], algorithms based on integer programming [2] or mixed approach based on both integer and linear programming principles [11,12] are used. In [5], a genetic algorithm for generating patterns is described. In [13], an approach based on metaheuristics is presented, in which the key idea is the construction of a pool of patterns for each given observation of the training set. Logical analysis of data is used in many application areas, such as cancer diagnosis and coronary risk prediction [2,10,11,14], credit risk rating [11,15–17], assessment of the potential of an economy to attract foreign direct investment [18], predicting the number of airline passengers [19], fault prognosis and anomaly detection [20–23], and others.

Thus, in traditional approaches to the logical analysis of data with homogeneous patterns, each pattern covers part of the observations of the target class and no observations of the opposite class. Otherwise, the homogeneity constraint is transformed into a relaxed non-coverage constraint related to a number or ratio of observations of the opposite class. Thus, such optimization model is not symmetric in the sense that the problem is focused on the number of covered observations of the target class while the coverage of the opposite class is considered as a constraint set at a certain level. Such an approach with the fuzzy patterns remains in the domain of the single-criterion optimization.

Recently, fuzzy logic theory has been widely developed in research. As mentioned in the literature [13], a certain degree of fuzziness seems to improve the robustness of the classification algorithm. In a fuzzy classification system, an object can be classified by applying a set of fuzzy rules based on its attributes. To build a fuzzy classification system, the most difficult task is to find a set of fuzzy rules pertaining to the specific classification problem [24]. To extract fuzzy rules, a neural net was proposed in several studies [25–27]. On the other hand, the decision tree induction method was used in [28–30]. In [30], a fuzzy decision tree approach was proposed, which can overcome the overfitting problem without pruning and can construct soft decision trees from large datasets. However, these methods were found to be suboptimal in solving certain types of problems [24]. In [13,24], a genetic algorithm for generating fuzzy rules was described. It was noted that they are very robust due to the global searching.

Fuzzy classification has practical applications in various fields. For instance, in [31], a fuzzy rule-based system for classification of diabetes was used. Authors in [32,33] have applied fuzzy theory in managing energy for electric vehicles. In [34], problems of a product processing plant related to the discovery of intrusions in a computer network were solved with use of a fuzzy classifier. In our study, we use fuzziness as a concept of partial patterns.

Both traditional and fuzzy approaches provide for finding both patterns covering the target class and patterns covering the opposite class. The methods for finding such patterns do not differ; in this sense, such approaches are symmetrical: the composition of patterns does not change from replacing the target class with the opposite one. At the same time, there is a significant difference between the requirement for maximum coverage and the requirement for purity of patterns.

When analyzing real data, pure patterns may be ineffective, and the concept of a pattern is extended to fuzzy patterns that cover some of the negative objects. This expansion occurs by relaxing the "empty intersection with negative objects" constraint. Thus, the aim of our study is to construct a classification model based on LAD principles, which does not impose a strict restriction nor relaxed constraint on the pattern coverage of the opposite class observations. Our model converts such a restriction (purity restriction) into an additional criterion. We formulate the pattern search problem as a two-criteria optimization problem: the maximum of covered observations of a certain class with the minimum of covered observations of the opposite class. Thus, our model has two competing criteria of the same scale, and the essence of solving the problem comes down to finding a balance between maximum coverage and purity of patterns. For this purpose, in this paper, we study the use of a multi-criteria genetic algorithm to search for Pareto-optimal fuzzy patterns. Our comparative results on medical test problems are not inferior to the results of commonly used machine learning algorithms in terms of accuracy.

The rest of the paper is organized as follows. In Section 2, we describe known and new methods implemented in our research. We provide the basic concepts of logical analysis of data (Section 2.1), an approach to formation of logical patterns (Section 2.2), a two-criteria optimization model for solving the pattern search problem, concepts of the evolutionary algorithm NSGA-II (Non-dominated Sorting Genetic Algorithm-II), developed to solve the multi-criteria optimization problem (Section 2.4). In Section 2.5, we discover the ability of evolutionary algorithms to solve the problem of generating logical patterns and describe modifications of the NSGA-II (Section 2.5). In Section 3, we present the results of solving two applied classification problems. In Sections 4 and 5, we discuss and shortly summarize the work.

2. An Evolutionary Algorithm for Pattern Generation

Several approaches which resemble in certain respects the general classification methodology of LAD can be distinguished [35]. For instance, in [36], a DNF learning technique was presented that captures certain aspects of LAD. Some machine learning approaches based on production or implication rules, derived from decision trees, such as C4.5 rules [37], or based on Rough Set theory, such as the Rough Set Exploration System [38]. The authors of [39] proposed the concept of emerging patterns in which the only admissible patterns are monotonically non-decreasing. The subgroup discovery algorithm described in [40] maximizes a measure of the coverage of patterns, which is discounted by their coverage of the opposite class. The algorithm presented in [35] maximizes the coverage of patterns while limiting their coverage of the opposite class. In [35], the authors introduced the concept of fuzzy patterns, considered in our paper.

2.1. Main Stages of Logical Analysis of Data

LAD is a data analysis methodology that integrates ideas and concepts from topics, such as optimization, combinatorics, and Boolean function theory [10,41]. The primary purpose of logical analysis of data is to identify functional logical patterns hidden in the data, suggesting the following stages [2,41].

Stage 1 (Binarization). Since the LAD relies on the apparatus of Boolean functions [41], this imposes a restriction on the analyzed data, namely, it requires the binary values of the features of objects. Naturally, in most real-life situations, the input data are not necessarily binary [2,12], and in general may not be numerical. It should be noted that in most cases, effective binarization [41] leads to the loss of some information [2,6].

Stage 2 (Feature extraction). The feature description of objects may contain redundant features, experimental noise as well as artifacts generated or associated with the binarization procedure [42,43]. Therefore, it is necessary to choose some reference set of features for further consideration.

Stage 3 (Pattern generation). Pattern generation is the central procedure for logical analysis of data [2]. At this stage, it is necessary to generate various patterns covering different areas of the feature space. However, these patterns should be of a sufficient level of quality, expressed in the requirements for the parameters of the pattern (for example, complexity or degree). To implement this stage, a specific criterion is selected as well as an optimization algorithm for constructing patterns that is relevant to the data under consideration [44].

Stage 4 (Constructing the classifier). When the patterns are formed, the classification of the new observation is carried out in the following way. An observation that satisfies the requirements of at least one positive pattern and does not satisfy the conditions of any negative patterns is classified as positive. The definition of belonging to a negative class is formulated similarly. In addition, it is required to determine how a decision will be made regarding controversial objects, for example, by voting on the generated patterns. The set of patterns formed at the previous step usually turns out to be too large and redundant for constructing a classifier, which leads to the problem of choosing a representative limited subset of patterns [2], such that it will provide a level of classification accuracy compared to using the complete set of rules. In addition, a decrease in the number of patterns makes it possible to increase the interpretability of the resulting classifier in the subject area [45].

Stage 5 (Validation). The last step of the LAD is not special and is inherent in other data mining methods. The degree of conformity of the model to the initial data should be assessed, and its practical value should be confirmed. In applied problems, the initial data reflect the complexity and variety of real processes and phenomena.

Stage 4 works fine on "ideal" data which means: reasonable amount of homogeneous data with no errors, no outliers (standalone observations which are very far from other ones), no gaps or inconsistency in data. When processing real data, we must take into account several issues [46]. Features may be heterogeneous (of different types and measured on different scales).

Data can be presented in a more complex form than a standard matrix of object features, for example, images, texts, or audio. Various data preprocessing methods are used to extract features. Another type of object description consists in pairwise comparison of objects instead of isolating and describing their features (featureless recognition [47]).

The number of objects may be significantly less than the number of features (data insufficiency). In systems with automatic collection and accumulation of data, the opposite problem arises (data redundancy). There are so much data that conventional methods process them exceptionally slowly. In addition, big amounts of data pose the problem of efficient storage.

The values of the features and the target variable (the label of class in the training sample) can be missed (gaps in data) or measured with errors (inaccuracy in data). Gross errors lead to the appearance of rare but large deviations (outliers). Data may be inconsistent which means that objects with the same feature description belong to different classes as a result of data inaccuracy.

Inconsistency and inaccuracy in data dramatically reduce the effectiveness of approaches based on homogeneous patterns. We have to apply fuzzy patterns in which a certain number of observations of the opposite class are allowed. At the same time, it is difficult to determine this threshold, since the level of data noise is usually unknown.

2.2. Formation of Logical Patterns

We restrict ourselves to considering the case of two classes: K^+ and K^-. Objects of K^+ class will be called positive sampling points, and objects of K^- class will be called negative. In addition, we assume that objects $X \in K^+ \cup K^-$ are described by k binary features, that is $x_i^{(j)} \in \{0,1\}\ \forall i,j$, where j is the object and i is the index of feature, $j = \overline{1,k}$.

LAD uses terms that are conjunctions of some literals (binary features x_i or their negations $1 - x_i$). We will say that a term C covers an object X if $C(X) = 1$. A logical positive pattern (or simply a pattern) is a term that covers positive objects and does not cover negative objects (or covers a limited number of negative objects). The concept of a negative pattern is introduced in a similar way.

Choose an object $a \in K^+$, assuming that $a = (a_1, \ldots, a_k)$ is the vector of feature values of this object. Denote by P_a a pattern covering the point a. The pattern is a set of feature values, which are fixed and equal for all the objects covered by the pattern. To distinguish fixed and unfixed features in pattern P_a, we introduce binary variables $Y^{(a)} = \left\{y_1^{(a)}, \ldots, y_k^{(a)}\right\}$ [48,49] as follows:

$$y_j^{(a)} = \begin{cases} 1, & \text{if } j\text{th feature is fixed in } P_a, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

A point $b \in K^+$ will be covered by a pattern P_a only if $y_j^{(a)} = 0 \;\; \forall j : \; b_j \neq a_j$. On the other hand, some point $c \in K^+$ will not be covered by the pattern P_a if $y_j = 1$ for at least one $j \in \{\overline{1,k}\}$ for which $c_j \neq a_j$.

It should be noted that any point $Y^{(a)} = \{y_1^{(a)}, \ldots, y_k^{(a)}\}$ corresponds to the subcube in the space of binary features $X = \{x_1,, \ldots, x_k\}$, which includes an object a.

It is natural to assume that the pattern can cover only a part of the observations from K^+. The more observations of positive class the pattern covers in comparison with observations of another type, the more informative it is [50]. Negative observation coverage is a pattern error.

Denote pattern P_a as a binary function of an object b: $P_a(b) = 1$ if object b is covered by pattern P_a, and 0 otherwise.

Noted constraints establish the minimum allowable clearance between the two classes. To improve the reliability of the classification, namely, its robustness to errors on class boundaries, constraints can be strengthened by increasing the value on the right side of the inequality.

Let us introduce the following notation: $Cov^+(P_a)$ is number of observations from K^+ for which the condition $P_a(b) = 1, \;\; b \in K^+$ is satisfied; $Cov^-(P_a)$ is number of observations from K^- for which the condition $P_a(c) = 1, \;\; c \in K^-$ is satisfied.

The pattern P_a is called "pure" if $Cov^-(P_a) = 0$. If $Cov^-(P_a) > 0$, then the pattern P_a is called "fuzzy" [35]. Obviously, among the pure patterns, the most valuable are the patterns with a large number of covered positive observations $Cov^+(P_a)$.

The number of covered positive observations $Cov^+(P_a)$ can be expressed as follows [2]:

$$Cov^+(P_a) = \sum_{b \in K^+} \prod_{j=\overline{1,k},\; b_j \neq a_j} (1 - y_j^{(a)}). \tag{2}$$

The condition that the positive pattern P_a should not cover any point of the negative class, which means the search for a pure pattern requires that for each observation $c \in K^-$, the variable $y_j^{(a)}$ takes the value 1 for at least one j for which $c_j \neq a_j$. Thus, a pure pattern is a solution to the problem of conditional Boolean optimization [2]:

$$\begin{gathered} \sum_{b \in K^+} \prod_{j=\overline{1,k},\; b_j \neq a_j} \left(1 - y_j^{(a)}\right) \to \max, \\ \sum_{j=\overline{1,k},\; c_j \neq a_j} y_j^{(a)} \geq 1 \quad \forall c \in K^-. \end{gathered} \tag{3}$$

Noted constraints establish the minimum allowable clearance between the two classes. To improve the reliability of the classification, namely, its robustness to errors on class boundaries, constraints can be strengthened by increasing the value on the right side of the inequality.

Since the properties of positive and negative patterns are completely symmetric, the procedure for finding negative patterns is similar.

2.3. Proposed Optimization Model

From the point of view of classification accuracy, pure patterns are preferable [10]. However, in the case of incomplete or inaccurate data, such patterns will have small coverage, which means, for many applications, the rejection of the search for pure patterns in favor of partial.

For the case of partial patterns, the constraint of the optimization problem $Cov^-(P_a) = 0$ transforms into the second objective function, which leads to the optimization problem simultaneously according to two criteria:

$$Cov^+(P_a) \to \max \quad \text{and} \quad Cov^-(P_a) \to \min\,. \tag{4}$$

The least suitable are those patterns that either cover too few observations or cover positive and negative observations in approximately the same proportion. The contradictions between these conflicting criteria can be resolved by transferring the second objective function to the category of constraints through establishing a certain admissible number of covered negative observations. In addition, multi-criteria optimization methods [51] can be applied, which construct an approximation of the Pareto front [52].

In addition, when searching for patterns, it is worth considering the degree of the pattern—the number of fixed signs of this pattern. It is easy to establish that there is an inverse dependence between the pattern degree and the number of covered observations (both positive and negative), so the pattern degree should not be too large [53].

The search for simpler patterns has well-founded prerequisites. Firstly, such patterns are better interpreted and understandable during decision-making. Secondly, it is often believed that simpler patterns have better generalization ability, and their use leads to better recognition accuracy [53]. The use of simple and short patterns leads to a decrease in the number of uncovered positive observations, but at the same time, shorter patterns can increase the number of covered negative observations. A natural way to reduce the number of false positives is to form more selective observations. This is achieved by reducing the size of the pattern determining subcube [48,54].

2.4. Algorithm NSGA-II

The NSGA-II [55] refers to evolutionary algorithms developed for solving the multi-criteria optimization problem that implements the approximation of the Pareto set. It is based on three key components: fast non-dominated sorting, estimation of the solutions location density, and crowded-comparison operator. In the original algorithm, solutions are encoded as vectors of real numbers, and the objective functions are assumed to be real-valued.

2.4.1. Fast Non-Dominated Sorting Approach

The basis of multi-criteria optimization is the selection of Pareto fronts [52] of different levels in the population of solutions. According to the intuitive approach, to isolate the non-dominated front, it is necessary to compare each solution with any other in the population for determining a set of non-dominated solutions.

For problem (4), the solutions are patterns, and a solution P_i is non-dominated in a population of N solutions $P_1, \ldots, P_N$ if $\nexists j \in \overline{1, N}$:

$$\big(Cov^+(P_j) \geq Cov^+(P_i) \text{ and } Cov^-(P_j) < Cov^-(P_i)\big)$$

$$\text{or } \big(Cov^+(P_j) > Cov^+(P_i) \text{ and } Cov^-(P_j) \leq Cov^-(P_i)\big).$$

After defining the first front, it is necessary to exclude representatives of this front from consideration and re-define the non-dominated front. The procedure is repeated until each solution is attributed to some front.

A more computationally efficient approach assumes for each solution P_j to keep track of the number of solutions n_{P_j} that dominate P_j, as well as a set of solutions S_{P_j} dominated by P_j.

Thus, all decisions in the first non-dominated front will have a dominance number equal to zero. For every solution P_j satisfying $n_{P_j} = 0$, we visit each member q of its set S_{P_j} and decrease its dominance number n_q by 1. Moreover, if for any member, the dominance number becomes zero, we put it in a separate list Q. Bypassing all the solutions of the first front, we find that the list Q contains solutions belonging to the second front. We repeat the procedure for these solutions similarly, looking through the dominated sets of solutions and reducing their dominance number until we identify all the fronts.

2.4.2. Diversity Maintenance

One of the problems of evolutionary algorithms is maintaining the diversity of the population [46,56,57]. Eremeev in [58] described the mutation genetic operator as the essential procedure that guarantees population diversity. Along with the convergence to the Pareto optimal set to solve the problem of multi-criteria optimization, the evolutionary algorithm must also support a good distribution of solutions, preferably evenly covering as much of the optimal front as possible [59–62].

Early versions of the NSGA used the well-known fitness-sharing approach to prevent the concentration of solutions in specific areas of the search space and maintain stable population diversity. However, the proximity parameter ς_{share} has a significant effect on the efficiency of maintaining a wide distribution of the population. This parameter determines the degree of redistribution of fitness between individuals [55] and is directly related to the distance metric chosen to calculate the measure of proximity between two members of the population. The parameter ς_{share} denotes the largest distance within which any two solutions share each other's suitability. The user usually sets this parameter, which entails apparent difficulties in making a reasonable choice.

In the NSGA-II [55], a different approach is used based on the crowded comparison. Its indisputable advantage is the absence of parameters set by the user. The critical components of the approach are the density estimate and the crowded-comparison operator.

To estimate the density of solutions concerning the chosen solution, along each direction in the criteria space, two nearest solutions are found on both sides of the chosen solution. The distance between them is determined as the difference between the values of a different criterion. An estimate of the density of solutions near the selected point will be the average of the calculated distances, called the crowding distance. From the geometric point of view, it is possible to estimate the density by calculating the perimeter of the hypercube formed by the nearest neighboring solutions as vertices.

Figure 1 shows a graphical interpretation of the above approach for the case of our two objective functions (4). Solutions denoted as p_{i+1} and $i - 1$, which are nearest to the ith solutions and belong to the first front (filled dots), represent the vertices of the outlined rectangle (dotted line). Crowding distance, in this case, can be defined as the average length of the edges of the rectangle.

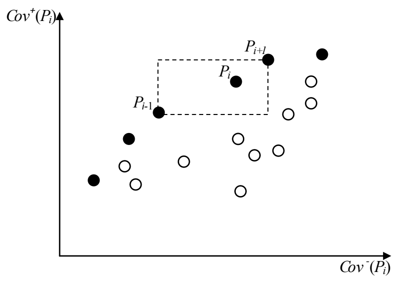

Figure 1. Density estimation of solutions belonging to the front for problem (4).

Calculating the crowding distance requires sorting the individuals in the population according to each objective function in ascending order of the value of this objective function. For individuals with the boundary value of the objective function (maximum or minimum), the crowding distance is assigned equal to infinity. All other intermediate solutions are assigned a distance value equal to the absolute value of the difference between the values of the functions of two neighboring solutions. This calculation continues for all objective functions. The final value of the crowding distance is calculated as the average of the individual values of the distances corresponding to each objective function. Pre-normalization of objective functions is recommended.

After all individuals in the population have been assigned to an estimate of the crowding distance, we can compare the solutions in terms of their degree of closeness to other solutions. A smaller value for the crowding distance indicates a higher density of solutions relative to the selected point. The density estimate is used in the crowded-comparison operator described below.

Crowded-Comparison Operator ($\prec_n$) directs the evolutionary process of population transformation towards a fairly uniform distribution along the Pareto front.

We assume that each individual in the population has two attributes:

1. Rank of dominance (front rank) i_{rank};
2. Crowding distance $i_{distance}$.

Then, the partial ordering is defined as follows. Individual i is preferred over individual j if the following conditions are met:

$$i \prec_n j \iff (i_{rank} < j_{rank}) \bigvee \left(i_{rank} = j_{rank} \bigwedge i_{distance} > j_{distance}\right). \tag{5}$$

Such ordering means that we prefer a solution with a lower (close to the first, which means the optimal front) rank between two solutions with different ranks. Otherwise, if both solutions are located in the same front, we prefer a solution located in areas where solutions are less crowded.

2.4.3. Basic Procedure of the NSGA-II

In this case, solving linear optimization problem is rather simple. Initially, the parent population P_0 is randomly created and sorted based on the dominance principle. Thus, the greater the suitability of an individual, the lower the value of its rank. Then, traditional genetic operators are applied: binary tournament selection, crossover, mutation, creating a child population Q_0 of a specified size N. Since elitism is introduced by comparing the current child population with the parent population, the procedure differs for the first generation from the repeated one.

Let the tth iteration of the algorithm be executed, at which the parent population P_t generated the child population Q_t. A joint population $R_t = P_t \cup Q_t$ is then sorted according to the dominance principle. Thus, decisions that belong to the first front $\mathcal{F}_1$ and are not dominated should have a better chance of moving to the next generation population, which is ensured by the implementation of the elitism principle. If the first front $\mathcal{F}_1$ includes less than N members, then all members of this front move to the next population P_{t+1}. The remaining members of the population P_{t+1} are selected from subsequent fronts in the order of their ranking. That is, the front $\mathcal{F}_2$ is included in the new parent population P_{t+1}, then the front $\mathcal{F}_3$, and so on. This procedure continues until the inclusion of the next front leads to an excess of the population size N. Let the front $\mathcal{F}_l$ no longer be included in the new population as a whole. To select members of the front $\mathcal{F}_l$ who will be included in the next generation, we sort the solutions of this front using the crowded-comparison operator and select the best solutions to supplement the population P_{t+1}. Now, a new population P_{t+1} of size N can be used to apply selection, crossover, and mutation operators. It is important to note that the tournament selection binary operator is still used, but now, it is based on the crowded-comparison operator $\prec_n$, to use which, in addition to determining the rank, it is necessary to calculate the crowding distance for each member of the population P_{t+1}.

The schematic procedure of the NSGA-II algorithm originally proposed in [55] is shown in Figure 2.

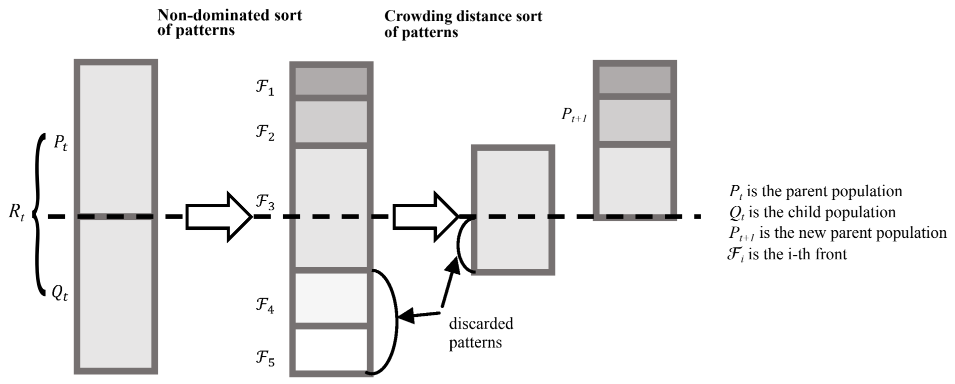
Non-dominated sort of patterns
Crowding distance sort of patterns
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathcal{F}_3$
$\mathcal{F}_4$
$\mathcal{F}_5$
P_t
R_t
Q_t
P_{t+1}
discarded patterns
P_t is the parent population
Q_t is the child population
P_{t+1} is the new parent population
$\mathcal{F}_i$ is the i-th front

Figure 2. Next-generation transition procedure.

Thus, a successful distribution of solutions within one front is realized by using the crowded-comparison procedure, which is also used in the individuals selection. Since the solutions compete with their crowding distances (a measure of the solution's density in a neighborhood), the algorithm does not require any additional parameter determining the size of niches in the search space. The proposed crowding distance is calculated in the function space, but it can be implemented in the parameter space if necessary.

2.5. Our Approach: An Evolutionary Algorithm for Pattern Generation

The pattern P_a is determined by baseline observation $a = \{x_1, x_2, \ldots, x_p\}$ and values of control variables $Y = \left\{y_1^{(a)}, \ldots, y_p^{(a)}\right\}$ that are binary values. Thus, the solution to the problem of pattern generation is a set of points in the space of control variables that approximate the Pareto front for a given base observation.

The posed problem of finding logical patterns determines the binary representation of solutions in the form of binary strings, rather than real variables, as was postulated in the original NSGA-II. The binary representation of the solution also determined the list of available crossover and mutation operators, among which uniform crossover and mutation by gene inversion were chosen [63,64].

In addition, the crowding distance in space "the number of covered observations of the base class" requires a different interpretation—"the number of covered observations of the class other than the base". Firstly, uniform coverage of the Pareto front is not required, since a preferable area is an area with a smaller scope of observations of the opposite class. Figure 3 illustrates this position: points of one Pareto front are colored according to their preference, with white color meaning less preference.

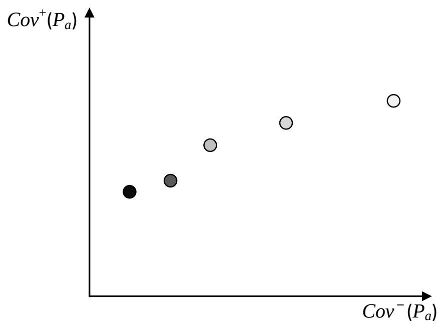

Figure 3. Illustration of a typical first Pareto front for logical patterns.

Taking into account these preconditions, the crowding distance was replaced by one of the heuristic definitions of informativity [3]:

$$I = \sqrt{Cov^{+}(P_a)} - \sqrt{Cov^{-}(P_a)}. \tag{6}$$

Similar to the crowding distance, the more preferable the individual, the greater the informative value.

Special attention is paid to the formation of the initial set of individuals. The usual approach is a uniform discrete distribution for values 0 and 1. However, based on the nature of the problem, an equal probability of dropping out 0 and 1 will mean, on average, the fixation of half of the features in the initial population, which gives rise to overly selective patterns. Therefore, the discrete distribution of values in the original population is defined differently: dropout 1 with probability p, and dropout 0 with probability $1 - p$. The value p will be the hyperparameter of the genetic algorithm.

Figure 4 shows a diagram of the developed approach to construct a classifier using the NSGA-II algorithm for pattern generation. The NSGA-II algorithm is run independently for each baseline observation. The launch result is a set of patterns of the first front. The sets of patterns obtained for each baseline observation were combined into one complete set of patterns, which can be reduced using the selection procedure [2,65]. When recognizing control (or new) observations, the decision about the class is made by balanced voting of patterns on the observation under consideration [8].

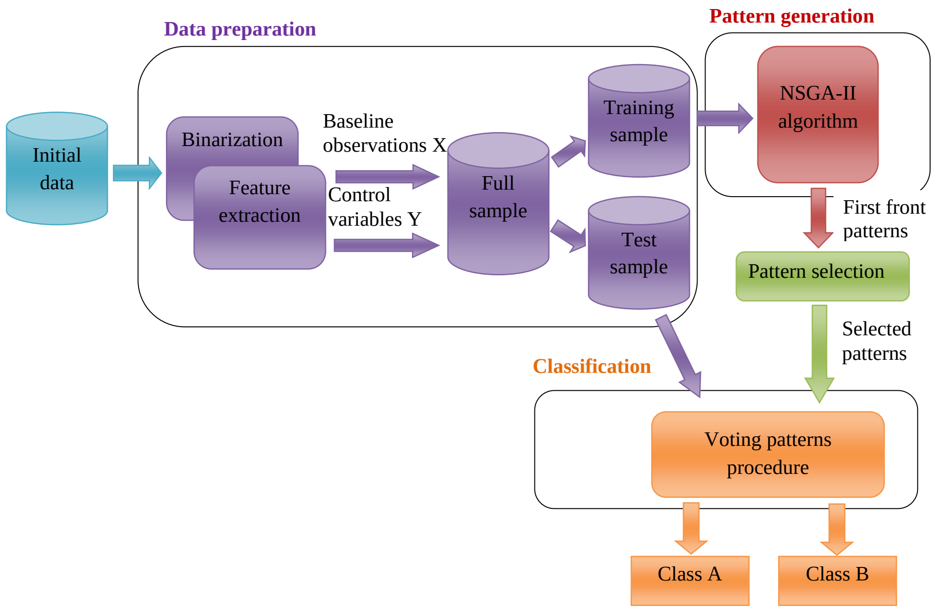

Figure 4. Scheme for constructing a classifier using the algorithm NSGA-II.

A detailed description of pattern generation procedure from Figure 4 is presented in a form of pseudocode (Algorithms 1 and 2).

Algorithm 1 Fast-non-dominated-sorting

```
Require: Evaluated population $P = \{P_1, \dots, P_N\}$, number of solutions in population $N$.
1: for each $P_i$, $i \in \{\overline{1, N}\}$ do
2:   for each $P_j$, $j \in \{\overline{1, N}\}$ do
3:     if $P_i$ dominates $P_j$ ($P_i \prec P_j$) then
4:       increase the set of solutions which the current solution dominates:
         $S_P i \leftarrow S_P i \cup \{P_j\}$
5:     else
6:       if $P_j$ dominates $P_i$ ($P_j \prec P_i$) then
7:         increase the number of solutions which dominate the current solution:
           $n_P i \leftarrow n_P i + 1$
8:       end if
9:     end if
10:  end for
11:  if no solution dominates $P_i$, $(n_P i = 0)$ then
12:    $P_i$ is a member of the first front: $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{P_i\}$
13:  end if
14: end for
15: $t \leftarrow 1$
16: while $\mathcal{F}_t \neq \emptyset$ do
17:   $\mathcal{H} \leftarrow \emptyset$
18:   for each $P_i \in \mathcal{F}_t$ do
19:     for each $P_j \in S_P i$ do
20:       $n_P j \leftarrow n_P j - 1$
21:       if $n_P j = 0$ then
22:         $\mathcal{H} \leftarrow \mathcal{H} \cup \{P_j\}$
23:       end if
24:     end for
25:   end for
26:   $t \leftarrow t + 1$
27:   $\mathcal{F}_t \leftarrow \mathcal{H}$
28: end while
29: return a list of the non-dominated fronts $\mathcal{F}$
```

NSGA-II is one of the popular multiobjective optimization algorithms. It is usually assumed that each individual in the population represents a separate solution to the problem. Thus, the population is a set of non-dominated solutions, from which one or more solutions can then be selected, depending on which criteria are given the highest priority. A distinctive feature of the proposed algorithm is, among other things, that the solution to the problem is the entire population, in which individual members represent individual patterns.

Algorithm 2 An evolutionary algorithm for pattern generation

```
Require: The set of baseline observations $X$ and values of control variables $Y$
 1: Create a random parent population $P_0$ of size $N$ from $X$ and $Y$
 2: $\mathcal{F} \leftarrow Fast-nondominated-sort(P_0)$
 3: Create a child population of size $N$ using selection, crossover and mutation procedures
    $Q_0 \leftarrow Child(P_0)$
 4: $t \leftarrow 0$
 5: while termination criteria not met do
 6:     Combine parent and child populations $R_t \leftarrow P_t \cup Q_t$. The size of population $R_t$ is
        $2N$
 7:     $\mathcal{F} \leftarrow Fast-nondominated-sort(R_t)$
 8:     while $|P_{t+1}| \geq N$ do
 9:         $I \leftarrow \sqrt{Cov^{+}(P_{t+1})} - \sqrt{Cov^{-}(P_{t+1})}$
10:         $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$
11:     end while
12:     Sort $P_{t+1}$ in descending order
13:     $P_{t+1} \leftarrow thefirstNelementsfromP_{t+1}$
14:     Create a child population using selection, crossover and mutation procedures $Q_{t+1} \leftarrow$
        $Child(P_{t+1})$
15:     $t \leftarrow t+1$
16: end while
17: return $P_{t+1}$
```

3. Application of the Proposed Method to Problems in Healthcare

The proposed approach relates to interpretable machine learning methods. Therefore, experimental studies were carried out on problems in which the interpretability of the recognition model and the possibility of a clear explanation of the solution proposed by the classifier are of great importance. The results are shown on two datasets: breast cancer diagnosis (the dataset from the UCI repository [66]) and predicting complications of myocardial infarction (regional data [67]).

3.1. Breast Cancer Diagnosis

The problem of diagnosing breast cancer on a sample collected in Wisconsin is considered (Breast Cancer Wisconsin, BCW) [66]. Since the attributes in the data take on numeric (integer) values, their binarization is necessary, that is, the transition to new binary features. Threshold-based binarization is used. Based on the original value x, a new binary variable x_t can be constructed as follows:

$$x_t = \begin{cases} 1, & \text{if } x \geq t, \\ 0, & \text{if } x < t, \end{cases} \tag{7}$$

where t is a cut point (threshold value).

As a result of executing the binarization procedure, 72 binary attributes were obtained from 9 initial attributes. The original dataset is divided in the ratio of 70% and 30% into training and test samples (in a random way), which results in 478 and 205 observations, respectively. To search for logical patterns, objects of the training sample (both positive and negative classes) consistently act as a baseline observation. For each baseline observation, the NSGA-II algorithm is run independently with the following parameters:

1. Parent population size: 20;
2. Descendant population size: 20;
3. Number of generations: 10;
4. Probability of mutation (for each decision variable): 5%;
5. Cross type: uniform.

The final set of patterns includes patterns of the first front from the last population of solutions. The first front can contain one or several patterns. Thus, the final set of patterns is completed based on covering the observations of the training sample. Therefore, each observation will be covered by at least one pattern from this set.

A series of experiments are carried out with a different probability of dropping one when initializing control variables in the first population of the genetic algorithm. The following probabilities were used: $p \in \{0.5, 0.3, 0.1\}$. A complete set of patterns was independently constructed for each probability. This set included the patterns of the first fronts obtained using the NSGA-II algorithm with the sequential acceptance of the observations of the training sample as a baseline observation. The resulting complete set of patterns are shown in Figure 5 in the coordinates of the number of covered observations of its own and the opposite class.

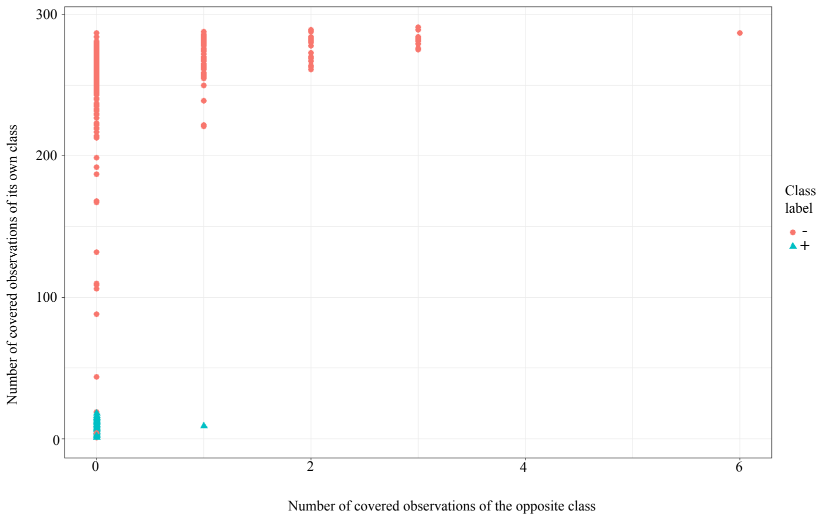

Figure 5. Aggregate scatter diagram of the detected patterns with $p = 0.5$.

Figure 6 shows the resulting full set of patterns for the value $p = 0.3$.

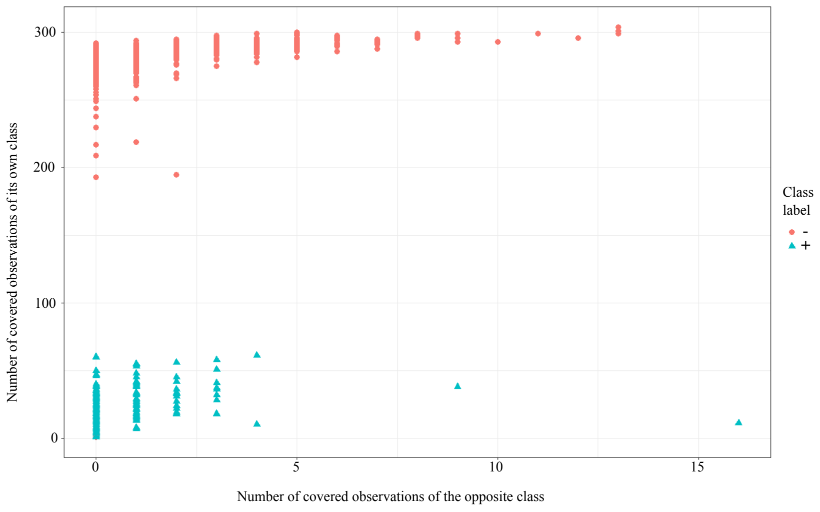

Figure 6. Aggregate scatter diagram of the detected patterns with $p = 0.3$.

Figure 7 shows the resulting full set of patterns for the value $p = 0.1$.

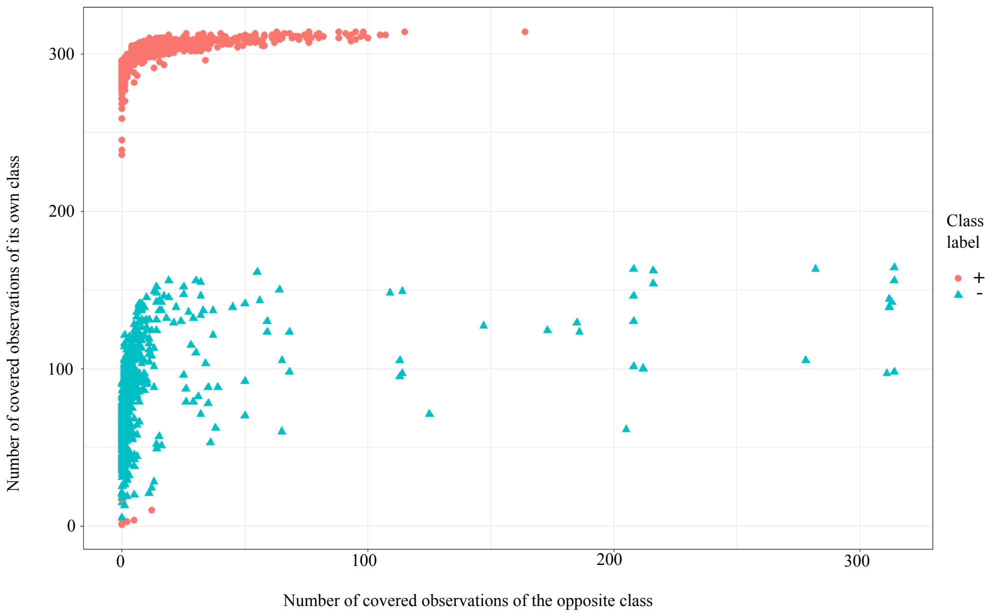

Figure 7. Aggregate scatter diagram of the detected patterns with $p = 0.1$.

Table 1 shows the distribution of patterns by power. Power is understood as the number of covered observations of its class included in the training sample. Data on the aggregate set of patterns are given without considering their class affiliation.

Table 1. Distribution of patterns in terms of power.

	Probability of Dropping 1 on Initialization		
The Number of Covered Observations of The Same Class on The Training Sample	***p* = 0.5**	***p* = 0.3**	***p* = 0.1**
1	1240	200	20
from 1 to 5	394	113	4
from 5 to 100	106	268	545
from 100 to 300	465	974	1770
above 300	0	3	962
Total number of patterns detected	2205	1558	3301

In the case of equally probable dropping of zero and one during initialization, a large number of trivial patterns are found, that is, patterns that cover only the basic observation and do not cover any more observations. These patterns are overly selective. With a decrease in the probability of dropping one during initialization, the selectivity of the obtained patterns decreases, thereby increasing their coverage. However, increased coverage affects the accuracy of initiated patterns, i.e., the number of observations from the opposite class.

The results of the classification of test observations compared to actual values are shown in Table 2. Symbols "+" and "−" denote the class labels introduced above. The symbol "?" means the impossibility of classification because none of the patterns is valid for observation.

Table 2. Confusion matrix for BCW problem.

***p* = 0.5**			***p* = 0.3**			***p* = 0.1**		
	Original Class			**Original Class**			**Original Class**	
Classifier Prediction	**−**	**+**	**Classifier Prediction**	**−**	**+**	**Classifier Prediction**	**−**	**+**
−	126	4	−	128	8	−	128	12
+	0	37	+	2	65	+	2	63
?	4	34	?	0	2	?	−	−
Accuracy	0.888		Accuracy	0.946		Accuracy	0.932	

Thus, a significant influence of the probability of dropping one is established during the initialization of control variables in the first population of the multi-criteria genetic algorithm. This parameter affects the selectivity of the detected patterns. The equiprobable dropping of zero and one entails a significant proportion of baseline observations, for which only patterns that cover only the baseline observation and no other training observations are found.

3.2. Predicting the Complication of Myocardial Infarction

The problem of predicting the development of complications of myocardial infarction is considered [67]. Datasets are often significantly asymmetric: one of the classes significantly outnumbers the other in terms of the number of objects. In our case, a sample contains data on 1700 cases with an uneven division into classes: 170 cases with complications, and 1530 cases without. Each case in the initial sample is described by 127 attributes containing information about the history of each patient, the clinical picture of the myocardial infarction, electrographic and laboratory parameters, drug therapy, and the characteristics of the course of the disease in the first days of myocardial infarction. Data include the following types: textual data, integer values, rank scale values with known range, real and binary values. A more detailed description of the attributes is given in Table 3.

The problem of predicting atrial fibrillation (AF) is solved, assuming that it is described by the variable 116 (target variable, 1—the occurrence of atrial fibrillation, 0—its absence). According to the nature of the attributes, variables $94, 95, 97, 98, 104, 105, 107, 108$ were excluded since these values could be obtained after the occurrence of complications [67]. The exclusion of these attributes significantly complicates the forecasting task. Thus, the number of predictors from the initial data was 107 variables.

The properties of the data used are as follows:

1. A significant number of missing values;
2. "Asymmetry" of the sample: the number of patients with atrial fibrillation is only about 10% of the total;
3. The presence of different types of attributes;
4. The use of homogeneous patterns leads to overfitting (the formation of patterns with a large number of conditions and a very small coverage).

Two approaches to data processing were implemented when constructing the classifier, as described below.

Table 3. Description of sample attributes for myocardial infarction problem.

Attribute Number	Description	Value Type
1	Full name of the patient (meta attribute)	text
2	Age	integer
3	Gender	binary
4–34	The presence or absence of various pathologies in the cardiovascular, endocrine and respiratory systems before the development of myocardial infarction	rank, binary
35–38	Blood pressure according to the data of the cardiology team and the admission department	integer
39–44	Complications arising at the time of transportation of the patient to the clinic or at the time of hospitalization	binary
45–49	Depth and localization of cardiac muscle necrosis	rank, binary
50–75	ECG characteristics at the time of admission of the patient to the intensive care unit	binary
76–82	The use of drugs during fibrinolytic therapy	binary
83–86	Electrolyte shifts in the blood	binary, real
87–92	Laboratory characteristics of blood	real, rank
93–115	The course of the disease in the first days of myocardial infarction	rank, binary
116–127	The occurrence of complications of myocardial infarction or the outcome of this disease	rank, binary

3.2.1. First Approach to Data Preparation (Complete Data and Handling of Missing Values)

The data contained a significant number of missing values. Columns with more than 100 missing values were excluded (47 variables). The remaining variables contained rank scale values and binary values. Missing values in columns containing 100 or less gaps were filled with the mode value. Thus, the prepared dataset contained 60 variables (not including the target).

The binarization for the values presented on the rank scale was performed based on the following rules. Based on the original variable x, a new binary variable x_t was constructed as follows:

$$x_t = \begin{cases} 1, & \text{if } x = r, \\ 0, & \text{if } x \neq r, \end{cases} \tag{8}$$

where r is the value of the original variable $x, r \in R$, such that R is a known set of all possible values of a variable x. As a result of the binarization procedure, the total number of binary variables was 119.

The original dataset is divided in a ratio of 70% and 30% into training and test samples, that is 1190 and 510 observations, respectively. To search for patterns, the training sample's objects (both positive and negative classes) consistently act as a baseline observation. For each baseline observation, the NSGA-II algorithm is run independently with the following parameters:

1. Parent population size: 20;
2. Descendant population size: 20;
3. Number of generations: 10;
4. Probability of dropping one on initialization: 0.05;
5. Probability of mutation (for each decision variable): 0.5%;
6. Crossing type: uniform.

More resources are allocated for seeking positive class observations. The sizes of the parent and descendant populations are 50, and the number of generations was 100. As a result, a complete set of patterns was obtained, containing 1961 positive class patterns and 6148 negative class patterns. A classifier is built that decides on a new observation by a simple voting. The results of the classification of the test observations compared to the actual values are shown in Table 4. Symbols "+" and "−" denote the class labels introduced above.

Table 4. Confusion matrix for AF problem (first approach).

	Original Class	
Classifier Prediction	**−**	**+**
−	347	31
+	109	23
Accuracy	0.7255	

Thus, the classifier arising from data for which missing values were processed did not show an acceptable result of classifying positive class objects, which may be caused by deleting essential data due to missing values processing. Since logical patterns are usable for classifying data with missing values, a different approach to data preparation was used, which is described below.

3.2.2. Another Approach to Data Preparation (Reduced Sampling without Processing Missing Values)

The second approach to data preparation is to store the missing attribute values as in the original sample. The class imbalance problem is also solved by randomly selecting a subset of objects of the prevailing class, equal in cardinality to the set of objects of another class.

As a result, the original sample was reduced to 338 cases, with an equal number of instances belonging to each class. Observations are described by 106 variables (not including the target). On the reduced sample, 8 variables take either just the same value, or the same value and gaps, so these variables were excluded from consideration. Among the remaining variables, 16 are rank variables, 12 are real, and the other variables are binary. Rank variables are transformed into several new binary variables, whose number is determined by the number of allowed ranks. Based on each real variable, four new binary variables are built. The threshold values of these variables were divided by the range of values of the original variables into four equal parts. As a result of applying the binarization procedure, the total number of variables was 200 (excluding the target variable).

Data were divided into training and test samples in the ratio of 80% and 20%, respectively, which causes 270 and 68 observations, respectively, in the same number of observations of positive and negative classes. For each training set observation, the NSGA-II algorithm is run independently with the following parameters:

1. Parent population size: 20;
2. Descendant population size: 20;
3. Number of generations: 10;
4. Probability of dropping one on initialization: 0.05;
5. Probability of mutation (for each decision variable): 0.5%;
6. Crossing type: uniform.

As a result, the cumulative set of patterns for the first front includes 2259 rules for the positive class and 2403 for the negative class. The classifiers are built based on reduced sets with the selection according to informativeness (I) of revealed patterns. Simple voting of patterns classifies a new observation. If the attribute value fixed in the pattern is unknown in the new observation, we assume that this pattern does not cover this observation. The classification results for different threshold values of informativeness are shown in Table 5.

Table 5. Confusion matrix for AF problem (the second approach).

$I > 0$			$I > 1$			$I > 2$		
	Original Class			**Original Class**			**Original Class**	
Classifier Prediction	**−**	**+**	**Classifier Prediction**	**−**	**+**	**Classifier Prediction**	**−**	**+**
−	25	7	**−**	25	7	**−**	25	5
+	9	27	**+**	9	27	**+**	9	29
Accuracy	0.7647		Accuracy	0.7647		Accuracy	0.7941	
$I > 0$			$I > 1$			$I > 2$		
	Original Class			**Original Class**			**Original Class**	
Classifier Prediction	**−**	**+**	**Classifier Prediction**	**−**	**+**	**Classifier Prediction**	**−**	**+**
−	23	4	−	23	3	−	30	18
+	11	30	+	11	31	+	4	16
Accuracy	0.7794		Accuracy	0.7941		Accuracy	0.6765	

Thus, the greatest accuracy is achieved when selecting patterns with informativeness greater than 2 or greater than 4. Since the importance of correctly identifying positive class objects (true positive rate or sensitivity) is greater than a negative one (true negative rate or specificity), the best option is to select patterns with informativeness greater than 4, since the accuracy of classifying objects in the positive class is higher. The obtained classification accuracy exceeds the accuracy obtained in [44].

Let us explore some characteristics of the resulting patterns when choosing patterns based on $I \geq 4$. Table 6 shows the number of patterns as well as the degree distribution of these patterns, that is, the number of binary variables fixed in the pattern.

Table 6. Degree of the patterns.

	Amount	**Min**	**1st Quartile**	**Median**	**Mean**	**3rd Quartile**	**Max**
Positive patterns	96	3	6	7	7.802	9	16
Negative patterns	103	6	9	11	11.480	14	19
All patterns	199	3	7	9	9.704	12	19

Figure 8 shows the indicated sets of patterns in the coordinates of the number of covered observations of its class and the opposite class. Figure 8 is given for the training sample.

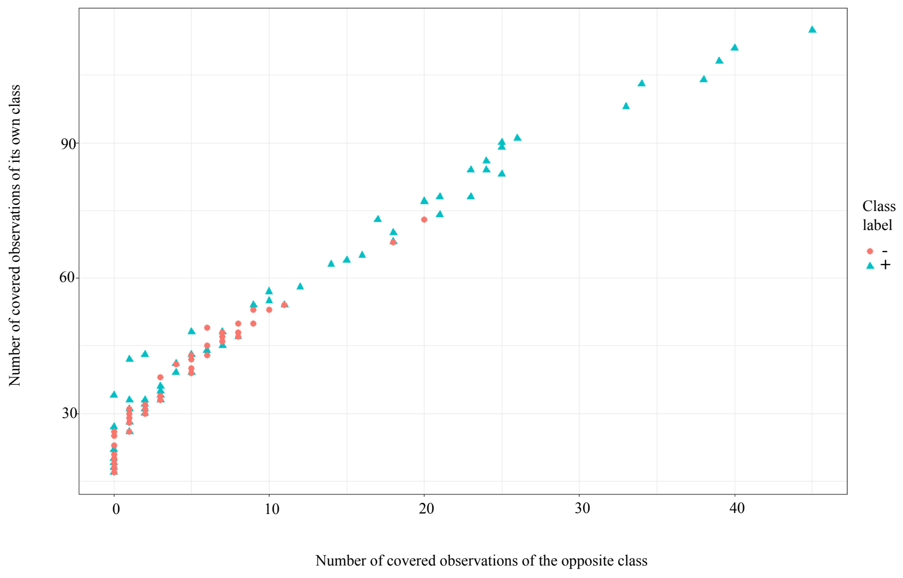

Figure 8. Cumulative scatter diagram of patterns on the training set.

Figure 9 shows the indicated sets of patterns in the coordinates of the number of covered observations of its class and the opposite class for the test sample.

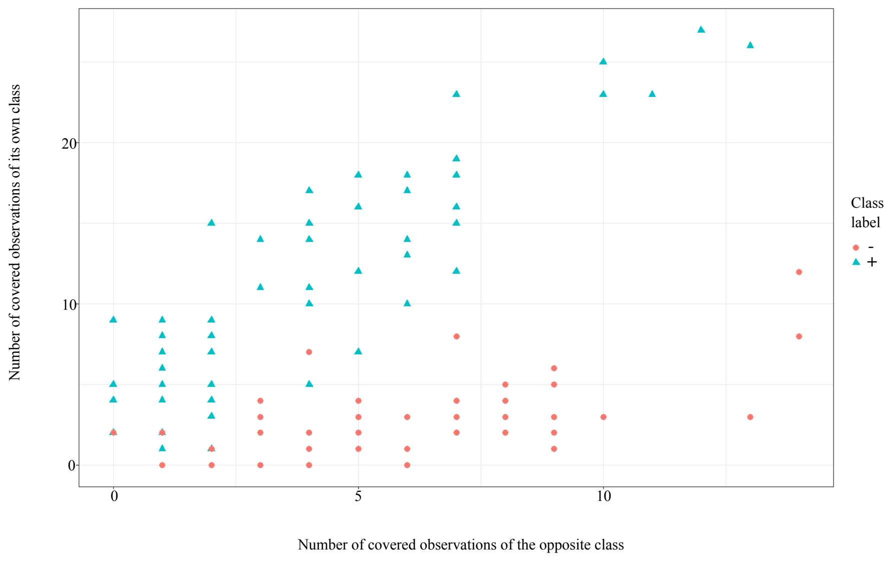

Figure 9. Cumulative scatter diagram of patterns on the test sample.

To evaluate the efficiency of the proposed approach, we made a comparative study with some widely used machine learning classification algorithms. We compared the proposed multi-criteria genetic algorithm (MGA-LAD) with Support Vector Machine (SVM), C4.5 Decision Trees (J48), Random Forest (RF), Multilayer Perceptron (MP), and Simple Logistic Regression (LR) methods. The tests were performed on the following datasets from UCI Machine Learning Repository [66]: Wisconsin breast cancer (BCW), Hepatitis (Hepatitis), Pima Indian diabetes (Pima), Congressional voting (Voting). Results using 10-fold cross-validation are presented in Table 7. Here, the "ML" column contains results for algorithms that give the highest accuracy among the commonly used machine learning algorithms. The best algorithm is given in parentheses. Another approach that has been used for comparison is logical analysis of data (LAD-WEKA in the WEKA package [68]) at various fixed fuzziness values φ (an upper bound on the number of points from another class that is covered by a pattern as a percentage of the total number of points covered by the pattern).

Table 7. Classification accuracy of the compared algorithms.

Dataset	ML	LAD, Fuzziness at Most φ					MGA-LAD
		φ = 0	φ = 0.05	φ = 0.1	φ = 0.15	φ = 0.2	
BCW	0.967 ± 0.01 (RF)	0.951 ± 0.03	0.953 ± 0.02	0.957 ± 0.03	0.966 ± 0.03	0.966 ± 0.03	0.965 ± 0.02
Hepatitis	0.858 ± 0.08 (SVM)	0.819 ± 0.09	0.826 ± 0.09	0.819 ± 0.09	0.806 ± 0.09	0.806 ± 0.08	0.847 ± 0.10
Pima	0.736 ± 0.03 (RF)	0.682 ± 0.02	0.687 ± 0.02	0.691 ± 0.02	0.687 ± 0.02	0.682 ± 0.02	0.749 ± 0.04
Voting	0.961 ± 0.01 (LR)	0.945 ± 0.03	0.952 ± 0.03	0.954 ± 0.03	0.954 ± 0.03	0.949 ± 0.03	0.973 ± 0.03

The value of the fuzziness parameter φ affects the size, coverage, and informativeness of the resulting patterns and ultimately affects the classification accuracy. In LAD, it is necessary to find many *a*-pattern based on different baseline observations. However, for each such pattern, the most appropriate fuzziness value may be different. Using a two-criteria model and the corresponding optimization algorithm allows us to find a set of Pareto optimal patterns without the need to fix the fuzziness value, which expands the possibilities of LAD and can improve the classification accuracy.

4. Discussion

In the previous section, we described the application of the proposed approach to solving practical medical classification problems.

The first dataset describes tissue characteristics to diagnose benign or malignant breast neoplasm. This dataset was studied many times before, for example [69,70], including the approach based on LAD [2]. For this example, we established a significant influence of the introduced hyperparameter of the genetic algorithm—the probability of dropping one during the initialization of the initial population, which is the fixation of the attribute's value in the pattern according to its value in the baseline observation. High values of this probability lead to low classification accuracy due to excessive selectivity of patterns and, accordingly, an increase in the number of objects with a refusal to determine the class membership.

The second dataset is the problem of predicting complications of myocardial infarction-atrial fibrillation. In this case, two approaches are implemented to handle missing values in the data. The first, typical for most data mining algorithms, is a combination of deleting values with missing values and filling them in. In this case, satisfactory classification results were not achieved when a complete dataset with a significantly larger presence of objects of one of the classes was used. In the second approach, the missing values are not preprocessed since the set of logical patterns as a whole has no restrictions in classifying observations with missing values. In addition, we use a reduced sample with an equal number of observations for each of the classes.

Homogeneous patterns in this dataset have small coverage, and using only homogeneous patterns leads to overfitting. Relaxation of homogeneity constraints requires adjusting the threshold (right-hand side of the constraint), which can be difficult since, when solving pattern finding problems, the best balance between coverage and homogeneity for a single pattern can be far from the best for another pattern (based on another baseline observation). The proposed approach simplifies the search for this balance since it considers many Pareto-optimal patterns. This approach prevents overfitting in contrast with using only homogeneous patterns or patterns with a given threshold for homogeneity. At the same time, the accuracy reaches the values obtained using an artificial neural network specially developed for these data [67]. The classification results are also comparable with the results of other works on this topic [71,72].

5. Conclusions

Logical analysis of data is a two-class learning method dealing with features that can be binarized. Logical analysis of data consists of several stages, and the formation of patterns is the most important. Finding pure patterns is a single-criteria optimization problem that consists of finding a pattern that covers as many positive observations as possible and does not cover negative observations. With a more general formulation, which allows making mistakes for a pattern, the problem turns into a search for a compromise between the completeness of the range of observations of some target class and the minimization of the coverage of observations that do not belong to the target class.

In this study, we used an approach to find patterns in the form of an approximation of the Pareto-optimal front and proposed evolutionary algorithms for solving such a problem due to their potential ability to cover the front. We modified the NSGA-II for pattern searching and tested it on several application problems from the repository.

Results of our work enabled us to find more informative patterns in the data, taking into account the coverage of objects of different classes. We compared our modified algorithm with commonly used machine learning algorithms on four classification problems. The results were comparable, and in some cases better than results of classical ML algorithms which do not meet the requirement of the interpretability of the result.

Our experiments discovered a significant influence of the probability of dropping one of the control variables of the initial population in the multi-criteria genetic algorithm. This parameter affects the selectivity of the selected patterns. So, the equal probability of zero and one entails a significant proportion of baseline observations, for which only patterns are found that cover only the baseline observation and no other observation of the training sample.

Since our two-criteria optimization model in a combination with the developed modification of the genetic algorithm does not require the number or ratio of observations of the opposite class to be pre-set. Thus, our approach is a more versatile tool of data analysis in this sense than known methods for the fuzzy patterns generation. The method considered in this paper could be useful for the classification tasks in, for instance, healthcare system, faults diagnosis and any problems in which the interpretability of results are of great importance.

Author Contributions: Conceptualization, I.S.M. and M.A.K.; methodology, I.S.M. and L.A.K.; software, M.A.K.; validation, I.S.M., M.K. and E.M.T.; formal analysis, P.S.S. and L.A.K.; investigation, I.S.M.; resources, I.S.M.; data curation, I.S.M. and M.A.K.; writing—original draft preparation, I.S.M., E.M.T. and L.A.K.; writing—review and editing, L.A.K., E.M.T., A.A.S. and P.S.S.; visualization, I.S.M., M.A.K. and A.A.S.; supervision, A.M.P., L.A.K. and P.S.S.; project administration, L.A.K. and A.M.P.; funding acquisition, L.A.K. and A.M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Ministry of Science and Higher Education of the Russian Federation, State Contract FEFE-2020-0013.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Initial data available at UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original), access date: 13 February 2022; https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications, access date: 13 February 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hammer, P.L. Partially defined boolean functions and cause-effect relationships. In *Proceedings of the International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*; University of Passau: Passau, Germany, 1986.
2. Hammer, P.L.; Bonates, T.O. Logical analysis of data: From combinatorial optimization to medical applications. *Ann. Oper. Res.* **2006**, *148*, 203–225. [CrossRef]
3. An, A.; Cercone, N. Rule Quality Measures for Rule Induction Systems: Description and Evaluation. *Comput. Intell.* **2001**, *17*, 409–424. [CrossRef]
4. Bruni, R.; Bianchi, G.; Dolente, C.; Leporelli, C. Logical Analysis of Data as a tool for the analysis of Probabilistic Discrete Choice Behavior. *Comput. Oper. Res.* **2019**, *106*, 191–201. [CrossRef]
5. Han, J.; Kim, N.; Yum, B.; Jeong, M. Pattern selection approaches for the logical analysis of data considering the outliers and the coverage of a pattern. *Expert Syst. Appl.* **2011**, *38*, 13857–13862. [CrossRef]
6. Boros, E.; Hammer, P.L.; Ibaraki, T.; Kogan, A.; Mayoraz, E.; Muchnik, I. An Implementation of Logical Analysis of Data. *IEEE T. Knowl. Data En.* **2000**, *12*, 292–306. [CrossRef]
7. Crama, Y.; Hammer, P.L.; Ibaraki, T. Cause-effect relationships and partially defined Boolean functions. *Ann. Oper. Res.* **1988**, *16*, 299–326. [CrossRef]
8. Lejeune, M.; Lozin, V.; Lozina, I.; Ragab, A.; Yacout, S. Recent advances in the theory and practice of Logical Analysis of Data. *Eur. J. Oper. Res.* **2019**, *275*, 1–15. [CrossRef]
9. Bain, T.; Ávila, J.; Subasi, E.; Subasi, M. Logical analysis of multiclass data with relaxed patterns. *Ann. Oper. Res.* **2020**, *287*, 11–35. [CrossRef]
10. Alexe, G.; Hammer, P.L. Spanned patterns for the logical analysis of data. *Discrete Appl. Math.* **2006**, *154*, 1039–1049. [CrossRef]
11. Guo, C.; Ryoo, H. On Pareto-Optimal Boolean Logical Patterns for Numerical Data. *Appl. Math. Comput.* **2021**, *403*, 126153. [CrossRef]
12. Lejeune, M.A. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Oper. Res.* **2012**, *60*, 1356–1372. [CrossRef]
13. Caserta, M.; Reiners, T. A pool-based pattern generation algorithm for logical analysis of data with automatic fine-tuning. *Eur. J. Oper. Res.* **2016**, *248*, 593–606. [CrossRef]
14. Alexe, S.; Blackstone, E.; Hammer, P.L.; Ishwaran, H.; Lauer, M.S.; Pothier Snader, C.E. Coronary risk prediction by logical analysis of data. *Ann. Oper. Res.* **2003**, *119*, 15–42. [CrossRef]
15. Hammer, P.L.; Kogan, A.; Lejeune, M.A. Modeling country risk ratings using partial orders. *Eur. J. Oper. Res.* **2006**, *175*, 836–859. [CrossRef]
16. Hammer, P.L.; Kogan, A.; Lejeune, M.A. A logical analysis of banks' financial strength ratings. *Expert Syst. Appl.* **2012**, *39*, 7808–7821. [CrossRef]
17. Rudin, C.; Shaposhnik, Y. Globally-Consistent Rule-Based Summary-Explanations for Machine Learning Models: Application to Credit-Risk Evaluation. Available online: https://ssrn.com/abstract=3395422 (accessed on 12 February 2022).
18. Bagchi, P.; Lejeune, M.A.; Alam, A. How supply competency affects FDI decisions: Some insights. *Int. J. Prod. Econ.* **2014**, *147 Pt B*, 239–251. [CrossRef]
19. Dupuis, C.; Gamache, M.; Page, J.-F. Logical analysis of data for estimating passenger show rates at Air Canada. *J. Air Transp. Manag.* **2012**, *18*, 78–81. [CrossRef]
20. Mortada, M.-A.; Yacout, S.; Lakis, A. Fault diagnosis in power transformers using multi-class logical analysis of data. *J. Intell. Manuf.* **2014**, *25*, 1429–1439. [CrossRef]
21. Das, T.K.; Adepu, S.; Zhou, J. Anomaly detection in Industrial Control Systems using Logical Analysis of Data. *Comput. Secur.* **2020**, *96*, 101935. [CrossRef]
22. Ragab, A.; El-Koujok, M.; Poulin, B.; Amazouz, M.; Yacout, S. Fault diagnosis in industrial chemical processes using interpretable patterns based on Logical Analysis of Data. *Expert Syst. Appl.* **2018**, *95*, 368–383. [CrossRef]
23. Jocelyn, S.; Ouali, M.-S.; Chinniah, Y. Estimation of probability of harm in safety of machinery using an investigation systemic approach and Logical Analysis of Data. *Saf. Sci.* **2018**, *105*, 32–45. [CrossRef]
24. Yuan, Y.; Zhuang, H. A genetic algorithm for generating fuzzy classification rules. *Fuzzy Sets Syst.* **1996**, *84*, 1–19. [CrossRef]
25. Hayashi, Y.; Imura, A. Fuzzy neural expert system with automated extraction of fuzzy if then rules from a trained neural network. In Proceedings of the First International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 3–5 December 1990; pp. 489–494.
26. Kosko, B. *Neural Networks and Fuzzy Systems*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1992.
27. Lin, C.-T.; Lee, C.S.G. Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Comput.* **1991**, *12*, 1320–1336. [CrossRef]
28. Weber, R. Automatic knowledge acquisition for fuzzy control applications. In *International Symposium on Fuzzy Systems*; Kyushu Institute of Technology: Fukuoka, Japan, 1992; pp. 9–12.
29. Yuan, Y.; Shaw, M.J. Induction of fuzzy decision trees. *Fuzzy Sets Syst.* **1995**, *69*, 125–139. [CrossRef]
30. Nguyen, H.S. Approximate Boolean Reasoning: Foundations and Applications in Data Mining. In *Transactions on Rough Sets V. Lecture Notes in Computer Science*; Peters, J.F., Skowron, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4100.

31. Aamir, K.M.; Sarfraz, L.; Ramzan, M.; Bilal, M.; Shafi, J.; Attique, M. A Fuzzy Rule-Based System for Classification of Diabetes. *Sensors* **2021**, *21*, 8095. [CrossRef] [PubMed]
32. Hussain, S.; Kim, Y.-S.; Thakur, S.; Breslin, J.G. Optimization of Waiting Time for Electric Vehicles Using a Fuzzy Inference System. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–12. [CrossRef]
33. Hussain, S.; Ahmed, M.A.; Kim, Y.-C. Efficient Power Management Algorithm Based on Fuzzy Logic Inference for Electric Vehicles Parking Lot. *IEEE Access* **2019**, *7*, 65467–65485. [CrossRef]
34. Kromer, P.; Platos, J.; Snasel, V.; Abraham, A. Fuzzy classification by evolutionary algorithms. In Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 313–318. [CrossRef]
35. Bonates, T.O.; Hammer, P.L.; Kogan, A. Maximum patterns in datasets. *Discrete Appl. Math.* **2008**, *156*, 846–861. [CrossRef]
36. Bshouty, N.H.; Eiron, N. Learning monotone DNF from a teacher that almost does not answer membership queries. *J. Mach. Learn. Res.* **2003**, *3*, 49–57.
37. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: Los Altos, CA, USA, 1993.
38. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*; Kluwer Academic Publishers: Norwell, MA, USA, 1992.
39. Dong, J.L. Efficient mining of emerging patterns: Discovering trends and differences. In Proceedings of the Fifth ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; ACM Press: New York, NY, USA, 1999; pp. 43–52.
40. Lavrac, N. Subgroup Discovery Techniques and Applications. In *Lecture Notes in Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3518, pp. 2–14.
41. Chikalov, I.; Lozin, V.; Lozina, I.; Moshkov, M.; Nguyen, H.S.; Skowron, A.; Zielosko, B. *Three Approaches to Data Analysis: Test Theory, Rough Sets and Logical Analysis of Data*; Springer: Berlin/Heidelberg, Germany, 2013. [CrossRef]
42. Yan, K.; Miao, D.; Guo, C.; Huang, C. Efficient feature selection for logical analysis of large-scale multi-class datasets. *J. Comb. Optim.* **2021**, *42*, 1–23. [CrossRef]
43. Bertolazzi, P.; Felici, G.; Festa, P.; Lancia, G. Logic classification and feature selection for biomedical data. *Comput. Math. Appl.* **2008**, *55*, 889–899. [CrossRef]
44. Kuzmich, R.; Stupina, A.; Korpacheva, L.; Ezhemanskaja, S.; Rouiga, I. The Modified Method of Logical Analysis Used for Solving Classification Problems. *Informatica* **2018**, *29*, 467–486. [CrossRef]
45. Letham, B.; Rudin, C.; McCormick, T.H.; Madigan, D. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.* **2015**, *9*, 1350–1371. [CrossRef]
46. Gasnikov, A. (Ed.) *Introduction to Mathematical Modeling of Traffic Flows*; MCCME: Moscow, Russia, 2013.
47. Too, J.; Abdullah, A.; Mohd, S.N.; Ali, N.; Tengku, Z.; Tengku, N.S. Featureless EMG Pattern Recognition Based on Convolutional Neural Network. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *14*, 1291–1297.
48. Masich, I.S.; Kazakovtsev, L.A.; Stupina, A.A. Optimization Models for Detection of Patterns in Data. In Proceedings of the School-Seminar on Optimization Problems and Their Applications (OPTA-SCL 2018), Omsk, Russia, 8–14 July 2018; pp. 264–275.
49. Kuzmich, R.I.; Masich, I.S.; Stupina, A.A.; Kazakovtsev, L.A. Algorithmic procedure for constructing the truncated basic set of characteristics in the method of logical analysis of data. In Proceedings of the 30th International Business Information Management Association Conference IBIMA 2017-Vision 2020: Sustainable Economic Development, Innovation Management, and Global Growth, Madrid, Spain, 8–9 November 2017; pp. 5592–5597.
50. Antamoshkin, A.N.; Masich, I.S.; Kuzmich, R.I. Heuristics and criteria for constructing logical patterns in data. In Proceedings of the International Scientific and Research Conference on Topical Issues in Aeronautics and Astronautics (Dedicated to the 55th Anniversary from the Foundation of SibSAU), Krasnoyarsk, Russia, 6–10 April 2015. [CrossRef]
51. Goh, C.-K.; Tan, K.C. *Evolutionary Multi-Objective Optimization in Uncertain Environments: Issues and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
52. Noghin, V.D. *Reduction of the Pareto Set. An Axiomatic Approach*; Springer: Berlin/Heidelberg, Germany, 2018. [CrossRef]
53. Hammer, P.L.; Kogan, A.; Simeone, B.; Szedmak, S. Pareto-optimal patterns in logical analysis of data. *Discrete Appl. Math.* **2004**, *144*, 79–102. [CrossRef]
54. Masich, I.S.; Kazakovtsev, L.A. A Branch-and-Bound Algorithm for a Pseudo-Boolean Optimization Problem with Black-Box Functions. *Facta Univ. Ser. Math. Inform.* **2018**, *33*, 337–360. [CrossRef]
55. Deb, K.; Pratar, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE T. Evolut. Comput.* **2002**, *6*, 182–197. [CrossRef]
56. Whitley, D. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, CA, USA, 4–7 June 1989; pp. 116–121.
57. Kazakovtsev, L.; Rozhnov, I.; Shkaberina, G. Increasing Population Variability in Parallel Genetic Algorithms with a Greedy Crossover for Large-Scale p-Median Problems. *IJAI* **2021**, *19*, 152–194.
58. Eremeev, A.V. Genetic Algorithm with Tournament Selection as a Local Search Method. *Discret. Anal. Oper. Res.* **2012**, *19*, 41–53. [CrossRef]
59. Veldhuizen, D.A.; Lamont, G.B. *Multi Objective Evolutionary Algorithm Research: A History and Analysis*; Tech. Rep.; Department of Electrical and Computer Engineering; Graduate School of Engineering; Air Force Institute of Technology; Wright-Patterson Air Force Base: Dayton, OH, USA, 1998.

60. Dai, C.; Wang, Y. A New Multiobjective Evolutionary Algorithm Based on Decomposition of the Objective Space for Multiobjective Optimization. *J. Appl. Math.* **2014**, *2014*, 906147. [CrossRef]
61. Jong, E.D.D.; Pollack, J.B. Multi-objective methods for tree size control. *Genet. Program. Evolv. Mach.* **2003**, *4*, 211–233. [CrossRef]
62. Liang, J.; Liu, Y.; Xue, Y. Preference-driven Pareto front exploitation for bloat control in genetic programming. *Appl. Soft Comput.* **2020**, *92*, 106254. [CrossRef]
63. Fogel, D. A parallel processing approach to a multiple travelling salesman problem using evolutionary programming. In Proceedings of the Fourth annual Symposium on Parallel Processing, Fullerton, CA, USA, 4–6 April 1990; pp. 318–326.
64. Fogel, D. An evolutionary approach to the travelling salesman problem. *Biol. Cybern.* **1988**, *60*, 139–144. [CrossRef]
65. Subasi, M.; Avila, J. *A New Approach to Select Significant Patterns in Logical Analysis of Data*; Rutcor Research Report; Rutgers University: New Brunswick, NJ, USA, 2012. [CrossRef]
66. UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original) (accessed on 13 February 2022).
67. UCI Machine Learning Repository: Myocardial Infarction Complications Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications (accessed on 13 February 2022).
68. Frank, E.; Hall, M.A.; Witten, I.H. *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Burlington, MA, USA, 2016.
69. Dada, E.G.; Ngene, C.; Daramola, C.Y. Performance Comparison of Machine Learning Techniques for Breast Cancer Detection. *NJEAS* **2017**, *6*, 1–8.
70. Sarmento, R. *Breast Cancer Wisconsin (Original) Data Set (Analysis with Statsframe ULTRA)*; Technical Report; University of Wisconsin Hospitals: Madison, WI, USA, 2019. [CrossRef]
71. Vizza, P.; Curcio, A.; Tradigo, G.; Indolfi, C.; Veltri, P. A Framework for the Atrial Fibrillation Prediction in Electrophysiological Studies. *Comput. Methods Programs Biomed.* **2015**, *120*, 65–76. [CrossRef]
72. Bashar, S.K.; Ding, E.; Walkey, A.; Mcmanus, D.; Chon, K. Atrial Fibrillation Prediction from Critically Ill Sepsis Patients. *Biosensors* **2021**, *11*, 269. [CrossRef]