



# Article Selected Data Mining Tools for Data Analysis in Distributed Environment

Mikhail Moshkov <sup>1</sup>, Beata Zielosko <sup>2,\*</sup> and Evans Teiko Tetteh <sup>3</sup>

- <sup>1</sup> Computer, Electrical and Mathematical Sciences and Engineering Division and Computational Bioscience Research Center, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
- <sup>2</sup> Institute of Computer Science, Faculty of Science and Technology, University of Silesia in Katowice, Będzińska 39, 41-200 Sosnowiec, Poland
- <sup>3</sup> Doctoral School, University of Silesia in Katowice, Bankowa 14, 40-007 Katowice, Poland
- \* Correspondence: beata.zielosko@us.edu.pl

Abstract: In this paper, we deal with distributed data represented either as a finite set  $\mathcal{T}$  of decision tables with equal sets of attributes or a finite set  $\mathcal{I}$  of information systems with equal sets of attributes. In the former case, we discuss a way to the study decision trees common to all tables from the set  $\mathcal{T}$ : building a decision table in which the set of decision trees coincides with the set of decision trees common to all tables from  $\mathcal{T}$ . We show when we can build such a decision table and how to build it in a polynomial time. If we have such a table, we can apply various decision tree learning algorithms to it. We extend the considered approach to the study of test (reducts) and decision rules common to all tables from  $\mathcal{T}$ . In the latter case, we discuss a way to study the association rules common to all information systems from the set  $\mathcal{I}$ : building a joint information system for which the set of true association rules that are realizable for a given row  $\rho$  and have a given attribute a on the right-hand side, and are realizable for the row  $\rho$ . We then show how to build a joint information system in a polynomial time. When we build such an information system, we can apply various association rule learning algorithms to it.

**Keywords:** distributed data; decision tables; information systems; decision trees; decision rules; tests; reducts; association rules

# 1. Introduction

Along with technological development, we are dealing with an increasing amount of data that must be processed and stored. The way they are processed depends on many factors, including the purpose of use and the type of data. One of the main goals is to extract knowledge from data, for example, by discovering patterns and relationships hidden in the data. Such knowledge can be presented by a set of decision rules, decision trees, or association rules. When a selection of features is required in order to find the most important and relevant ones, a test (reduct) is used. It is a (minimal) set of attributes that provides the same classification of objects as the whole input set of features.

An important element that influences the result of the chosen approach to extracting knowledge from data is their preparation. Pre-processing includes various algorithms, depending on the needs. These can be, for example, the imputation of missing attribute values, data normalization, or discretization. The type of method used depends on the goal and affects the subsequent stages of the data mining process. This phase is particularly difficult when we are dealing with distributed data that come from various data sources and appear in a different format, depending on the data owner [1].

One popular form of data representation is the tabular form, presented either as a decision table or as an information system. In the case of a distributed environment, such



Citation: Moshkov, M.; Zielosko, B.; Tetteh, E.T. Selected Data Mining Tools for Data Analysis in Distributed Environment. *Entropy* **2022**, *24*, 1401. https://doi.org/10.3390/e24101401

Academic Editors: Przemysław Juszczuk and Jan Kozak

Received: 2 August 2022 Accepted: 14 September 2022 Published: 1 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). data can be represented as a finite set of decision tables with the same decision attribute [2,3]. Generally, these decision tables can have different sets of conditional attributes. However, the consideration of the sets of decision tables with equal sets of attributes is of particular interest. Data can also be represented by information systems [4,5]. As for the case of decision tables, considering the sets of information systems with equal sets of attributes is of most interest to us. This paper consists of the two parts. In the first one, we deal with dispersed data represented by a finite set of decision tables with equal sets of attributes. In the second part, we deal with dispersed data represented by a finite set of attributes.

In the first part of the paper, we assume that we have a finite set  $\mathcal{T} = \{T_1, ..., T_k\}$  of decision tables with equal sets of attributes. Our aim is to create tools for the work with decision trees, rules, and tests (reducts) [4–6] that are common to all decision tables from  $\mathcal{T}$ .

There are different algorithms for the construction and optimization of decision trees for single decision tables [7–10]. To apply these algorithms to the set of decision tables  $\mathcal{T}$ , we need to build a single decision table (called a joint decision table for  $\mathcal{T}$ ) such that the set of decision trees for this table is equal to the set of common decision trees for all decision tables from  $\mathcal{T}$ . The situation is the same for decision rules and tests (reducts). In this paper, we show when we can build joint decision tables and how to build them in a polynomial time.

Note that in the case of dispersed decision tables with different sets of conditional attributes, instead of considering a joint decision table, we should study its lower and upper approximations, which leads to the investigation of NP-hard problems [2].

In the second part of the paper, we assume that we have a finite set  $\mathcal{I} = \{I_1, \ldots, I_k\}$  of information systems, in which columns are labeled with the same attributes  $a_1, \ldots, a_n$ . We fix a row  $\rho$  from one of the information systems from  $\mathcal{I}$  and an attribute  $a_j \in \{a_1, \ldots, a_n\}$ , and we consider the set  $Arules(\mathcal{I}, \rho, a_j)$  of association rules of the form  $(a_{i_1} = \sigma_1) \land \cdots \land (a_{i_m} = \sigma_m) \rightarrow (a_j = \sigma)$  that are true for each information system from  $\mathcal{I}$  and are realizable for the row  $\rho$  (i.e., such rule covers the row  $\rho$ ). Our aim is to create tools for the work with association rules from this set.

There are different algorithms for the construction and optimization of association rules for single information systems [11–16]. To apply these algorithms to the set of information systems  $\mathcal{I}$ , we need to build an information system J (called a joint information system for  $\mathcal{I}$ ,  $\rho$ , and  $a_j$ ) such that  $Arules(\{J\}, \rho, a_j) = Arules(\mathcal{I}, \rho, a_j)$ . In this paper, we show how to build joint information systems in a polynomial time.

The main contribution of this work is a proposed new methodology for working with distributed data, presented as a set of decision tables or a set of information systems. It is an interesting direction of research, especially in the areas of distributed data mining, data processing, and knowledge extraction from dispersed data sources. The proposed approach is different from the approaches described in the framework of distributed data mining (Section 2.1). Our methodology is based on the transformation of distributed data sources into the so-called joint tabular form of data, presented as a joint decision table or as a joint information system. An important element is that the obtained decision table or information system allows for the induction of decision rules, decision trees, reducts, or association rules common to the distributed data. Moreover, existing algorithms for their induction can be used.

The present paper is an extended version of two conference papers [17,18].

The rest of the paper is organized as follows. Section 2 presents some background information related to distributed data, decision trees and rules, tests, and reducts as well as association rules. In Section 3, we study distributed data represented as a finite set of decision tables, and in Section 4, we study distributed data represented as a finite set of information systems. Section 5 contains brief conclusions.

## 2. Background Information

In this section some basic information related to distributed data, decision trees and rules, tests, and reducts as well as association rules is presented.

# 2.1. Distributed Data

Technological development means that we are dealing with an increasing amount of data that can be heterogeneous, taking into account their format and location.

One of the popular solutions for processing and storing decentralized data are data warehouses [19,20]. They are used to store huge data sets. By using appropriate analytical tools that allow for the employment of data mining algorithms, it is possible to mine knowledge from data by analyzing trends, anomalies, or searching patterns. On this basis, business decisions are made regarding, for example, sales planning or marketing campaigns. In addition, data warehouses have ETL (Extraction, Trasformation, Loading) tools, which are designed to properly prepare data from heterogeneous sources and various locations.

Along with technological development and the necessity to process large amounts of distributed data, the field referred to as distributed data mining has been developing in recent years [21,22]. In this framework, different algorithms and approaches have been developed and proposed for classification, association mining, clustering, and other data mining tasks [23,24].

In this paper, a new methodology for working with distributed data is proposed. It is based on the idea of constructing one tabular form of data representation, i.e, a decision table or an information system for distributed sources, and then applying known algorithms for the induction of data mining tools, i.e., association and decision rules, decision trees, and reducts.

It should also be taken into account that distributed data mining techniques are more complex in comparison to centralized ones. The main issues which should be considered are: (i) heterogeneous data, i.e., local data sources can provide data with different formats and attributes with different domains; (ii) data fragmentation, i.e., local sources can be viewed as a horizontal or vertical fragmentation of the global data table, and therefore based on them, only part of the knowledge can be induced; (iii) data replication, i.e., replication provides better data availability, but on the other hand, it can make it difficult to ensure the consistency of distributed data; (iv) cost of communication in a distributed environment plays an important role; (v) security, privacy, and autonomy of local sources; (vi) integration results, i.e., discovered global interesting patterns and associations should be collected from local sources, and their utility should be verified globally.

Distributed data mining aims to analyze and process distributed data while taking into account resource constraints [25]. This task can be realized in the framework of a metalearning, multi-agent system, or based on grid. The multi-agent data mining environment inherits properties of agents as interoperability and performance aspects. Interoperability concerns working collaboratively with other agents in the entire system. Performance measures can be improved or impaired by the data distribution at the local level. The meta-learning system constitutes a learning method at the local level. Learning at the meta level is based on accumulating experience on the performance of multiple applications of a learning system. Data mining based on grid aims to create a distributed computing environment in order to enable local data sources to use computing resources on demand.

## 2.2. Data Mining Tools

Data mining is a complex process that allows for the performance of analyses and the acquisition of knowledge from data by using different methods, depending on the aim and kind of data. Among data mining tools, decision rules, decision trees, reducts, and association rules can be used. They can be considered as algorithms for solving different problems and also as classifiers used in the area of machine learning [26]. A short description can be found in the sections below.

## 2.2.1. Decision Rules

Decision rules are popular and an often used form of knowledge representation. In general, decision rules can be presented in the following form:

*IF* condition<sub>1</sub> 
$$\wedge \ldots \wedge$$
 condition<sub>k</sub> *THEN* conclusion. (1)

Conditions (pairs attribute = value) correspond to descriptors that are present in the premise part of the rule. Conclusion corresponds to the rule consequent part that present a class label. Rules presented in such a form can be considered as a compact form of knowledge representation. This form is simple and easily accessible from the point of view of understanding and interpreting knowledge represented by rules. Moreover, decision rules based on background knowledge can be employed in classification tasks, where a class label for a new object is assigned based on its conditions. Hence, decision rules can be applied in data mining tasks related to (i) knowledge representation and (ii) classification [27]. Taking into account these two perspectives, there are different measures used for rule evaluation and many different approaches for the induction of decision rules. The aim is to find patterns or regularities hidden in the data that are interesting and useful for users.

It should be noted that the minimization of length (number of conditions) and the maximization of support (which allows to discover major patterns in data) of decision rules are NP-hard problems [6,14]. The most part of approaches for construction of decision rules, with the exception of brute force, Boolean reasoning [28], and dynamic programming [6], cannot guarantee the construction of optimal rules, i.e., rules with minimum length or maximum support. Consequently, different heuristic approaches have been proposed in the literature [26,27,29,30]. Among them, greedy algorithms, genetic algorithms, ant colony optimization algorithms, approaches based on a sequential covering procedure, and many others can be mentioned.

# 2.2.2. Decision Trees

Decision trees are often used as classifiers, as a means of knowledge representation, and as algorithms. A decision tree learning algorithm approximates a target concept using a tree representation, where each internal node corresponds to an attribute, and each terminal node known as a leaf corresponds to a class label. The root node is at the top and leafs are at the bottom of a tree.

Most of the algorithms for decision tree induction use a greedy approach and a topdown, recursive, divide-and-conquer technique. In general, the algorithm for decision tree induction starts with the tree, which initially contains a single root node that is associated with the objects included in a data set. Then, the instances are recursively partitioned into smaller subsets according to a given splitting criterion. It indicates the attribute chosen as the test condition and how the instances should be distributed to the child nodes of the constructed tree. The creation and expansion of a node is finished when the stop criterion is satisfied, for example, when all the instances associated with the node in the divided data set have the same class label. However, there are also other criteria that allow for the expansion of a node to be stopped earlier even if corresponding assigned instances have different decisions.

An advantage of decision trees is that by reading a tree from root to leaves, a decision (class label) is proposed for a considered case (object); it is also possible to see the reasons for choosing a given decision. This feature is a very important element used in the domain of applications aimed at supporting decision making. In addition, based on the decision tree, decision rules can be obtained.

There are many algorithms for decision tree induction. The most popular are [8,9,31,32]: CART (Classification and Regression Trees), ID3 (Iterative Dichotomiser 3), C4.5 (improved version of the ID3 algorithm, where "C" shows that algorithm was written in C and the 4.5 specifics version of this algorithm), Sprint (Scalable PaRallelizable INduction of decision Trees), Chaid (Chi-square automatic interaction detection), and their many modifications.

There are also a variety of approaches based on meta-heuristics [33] such as genetic algorithms, simulated annealing, ant colony optimization, and many others. An important element during decision tree induction is selecting the best split, which allows for the partitioning of instances into two or more subsets that are associated with the nodes of the decision tree. Among the popular ones, measures based on entropy and the Gini index used in CART can be distinguished.

#### 2.2.3. Tests and Reducts

The construction of reducts and tests (super reducts) is closely connected with the feature selection area [34–36]. The aim of this domain is to select from the entire set of features only those attributes that are the most relevant while maintaining the descriptive and classification properties of the original feature space. Hence, this reduced set of attributes can be used instead of the entire attributes set for knowledge discovery. It is an important task, especially in areas where data sets contain a huge number of features, for example, in market basket analysis, stock trading, and sequence pattern discovery in bioinformatics.

Reduct is as an irreducible subset of features providing a satisfactory level of information about the considered target variable, which can be, for example, the accuracy of the classifier constructed based on the features contained in it. Therefore, from the classification point of view, a reduct can be interpreted as a minimal subset of attributes that has the same classification power as the entire set of features. Definitions for attribute reducts can be based on different criteria, for example, a reduct can also be considered as a minimal set of attributes that preserves the degree of dependency of the full set of attributes [37].

In the rough sets theory, where the construction of reducts constitutes one of the main research directions, decision super reduct (test) is defined as a subset of condition attributes that is sufficient for discerning any of the objects in a decision table with different class labels. A decision reduct is a test in the sense that each proper subset of this test is not a test for the considered problem.

Unfortunately, finding a reduct with minimum cardinality is an NP-hard problem. It is also known that the upper bound of a potential number of all reducts that can be found for a given dataset with *k* attributes is equal to  $\binom{k}{\lfloor k/2 \rfloor}$ . Taking into account that these issues represent high computational costs and complexity brought by the tasks of all reduct construction, different approaches and heuristics have been proposed for the construction of many reducts in some acceptable time. The popular ones are Boolean reasoning [28], genetic algorithms [38], greedy algorithms [39], fuzzy-rough approach, and others [14,40].

Based on the reduct constructed for a given decision table, decision rules can be induced from reduced sets of attributes. In this indirect method of rule induction, it is easy to see that the number of attributes which constitute a reduct is an important factor from the point of view of knowledge representation. Short reducts allow for the construction of short decision rules, which are more preferred from the point of view of understanding and interpretation by users.

## 2.2.4. Association Rules

Association rule mining is one of the key and interesting methods of data mining and knowledge discovery. It aims to extract co-occurrences of items as well as associations and patterns hidden in the data. One of the most popular applications of association rules is the market basket analysis, which finds associations between different items that customers place in their shopping baskets. Other areas include business fields involving decision making and effective marketing, medical diagnosis, stock trading, and others.

There are different types of association rules, for example: boolean association rules, which are used in market basket analysis; qualitative association rules [11], which are

$$X \to Y$$
, (2)

where *X* and *Y* are sets of items.

Two main quality measures of association rules are support and confidence [15]. Rules that satisfy minimum thresholds of these measures indicated by a user are called strong association rules.

It should be also noted that there are many algorithms for construction of association rules, however the process of mining of association rules consists of two main stages: (i) find all frequent itemsets, i.e., they occur at least as frequently as a predetermined minimum support threshold, and (ii) generate strong association rules from the frequent itemsets, i.e., rules that satisfy minimum support and minimum confidence thresholds. The most popular algorithm based on mining frequent itemset is Apriori [43]. However, many other approaches were proposed by researchers, for example, algorithms that use frequent pattern growth approach [44], vertical data format [45], hash based technique, partitioning the data and others [46].

One very important task in data mining is the classification process. In this framework, association rules also have an application. The associative classification task aims to find association rules that have only the class label in the consequent part of the rule and which satisfies the minimum support and the confidence thresholds, the so-called Class Association Rules. There are many methods for the construction of classifiers, which differ in the approaches used for mining association rules and their selection [47].

#### 3. Sets of Decision Tables

In this section, we deal with dispersed data represented as a finite set of decision tables with equal sets of attributes.

#### 3.1. Main Notions

A decision table *T* is a table filled with numbers from the set  $\omega = \{0, 1, 2, ...\}$  of nonnegative integers, in which columns are labeled with conditional attributes  $a_1, ..., a_n$  and each row is labeled with a decision that is a number from  $\omega$  (see Figure 1). We assume that equal rows in the table *T* are labeled with equal decisions, i.e., we consider only consistent decision tables. We associate the following problem with the table *T*: for a given row  $\rho$  of *T*, we should recognize the decision attached to  $\rho$  using values of the condition attributes from  $\{a_1, ..., a_n\}$  in this row. To this end, we can use decision trees, rules, and test (reducts).

A decision tree  $\Gamma$  over T is a finite directed tree with a root, in which each internal node is labeled with an attribute from the set  $\{a_1, \ldots, a_n\}$ , edges leaving this node are labeled with pairwise different numbers from  $\omega$ , and each leaf node is labeled with a decision from  $\omega$ . For a given row  $\rho = (\delta_1, \ldots, \delta_n)$ , the tree  $\Gamma$  work starts in the root of  $\Gamma$ . If the node under consideration is a leaf, then the number attached to this node is the result of the  $\Gamma$  work. Let the node under consideration be an internal node with an attribute  $a_i$  attached to it. If there is an edge that leaves the considered node and is labeled with  $\delta_i$ , then we pass along this edge. Otherwise, the decision tree  $\Gamma$  finishes its work without a result. We say that  $\Gamma$  is a decision tree for T if, for any row of T, the work of  $\Gamma$  finishes in a leaf that is labeled with the same decision as the considered row (see Figure 1). We denote with Trees(T) the set of decision trees for T.

Any decision rule over *T* can be represented in the following form:

$$(a_{i_1} = \sigma_1) \wedge \dots \wedge (a_{i_m} = \sigma_m) \to t \tag{3}$$

where  $a_{i_1}, \ldots, a_{i_m} \in \{a_1, \ldots, a_n\}$  and  $\sigma_1, \ldots, \sigma_m, t \in \omega$ . This rule is called realizable for a row  $\rho = (\delta_1, \ldots, \delta_n) \in \omega^n$  (it is possible that this row does not belong to *T*) if  $\delta_{i_1} = \sigma_1, \ldots, \delta_{i_m} = \sigma_m$ . This rule is called true for *T* if, for any row  $\rho'$  of *T*, such that rule (3) is realizable for  $\rho'$ , the row  $\rho'$  is labeled with the decision *t*. We say that (3) is a rule for *T* and  $\rho$  if this rule is true for *T* and realizable for  $\rho$  (see Figure 1). We denote with  $Rules(T, \rho)$  the set of decision rules for *T* and  $\rho$ . One can show that (3) is a rule for *T* and  $\rho$  if (i)  $\rho$  is labeled with the decision *t* if  $\rho$  belongs to *T*, and (ii) if each row  $\rho'$  of *T*, which is labeled with a decision different from *t*, is different from  $\rho$  on at least one attribute from the set  $\{a_{i_1}, \ldots, a_{i_m}\}$ .

A test for *T* is a subset of the set of conditional attributes  $\{a_1, ..., a_n\}$ , such that any two rows from *T* with different decisions are different on at least one attribute from this subset. A reduct for *T* is a test for *T*, for which each proper subset is not a test (see Figure 1). We denote with Tests(T) the set of tests for *T*.



**Figure 1.** Considered objects: (a) decision table  $T_0$ , (b) decision rule for  $T_0$  and row (1,0,0), (c) reduct for  $T_0$ , (d) decision tree for  $T_0$ .

Let  $\mathcal{T} = \{T_1, \ldots, T_k\}$  be a finite nonempty set of decision tables, in which columns are labeled with the same conditional attributes  $a_1, \ldots, a_n$ . Each decision table from this set is consistent, but different tables from  $\mathcal{T}$  can contain equal rows labeled with different decisions. Let  $\rho$  be a row of a decision table from  $\mathcal{T}$ . We denote  $Trees(\mathcal{T}) = \bigcap_{T_i \in \mathcal{T}} Trees(T_i)$ ,  $Rules(\mathcal{T}, \rho) = \bigcap_{T_i \in \mathcal{T}} Rules(T_i, \rho)$ , and  $Tests(\mathcal{T}) = \bigcap_{T_i \in \mathcal{T}} Tests(T_i)$ . In the next three sections, we will consider joint decision tables for these sets of common decision trees, rules, and tests (reducts) for  $\mathcal{T}$ .

# 3.2. Joint Decision Tables for Decision Trees

Let  $\mathcal{T} = \{T_1, \ldots, T_k\}$  be a set of decision tables, in which the columns are labeled with the attributes  $a_1, \ldots, a_n$ . The set of decision tables  $\mathcal{T}$  is called consistent if there are no two tables in  $\mathcal{T}$  containing equal rows labeled with different decisions.

First, we show that if the set  $\mathcal{T}$  is not consistent, then  $Trees(\mathcal{T}) = \emptyset$ . Since  $\mathcal{T}$  is not consistent, there exist two tables  $T_i$  and  $T_j$  in  $\mathcal{T}$  and a row  $\rho$ , such that  $\rho$  is a row of  $T_i$  labeled with a decision p,  $\rho$  is a row of  $T_j$  labeled with a decision q, and  $p \neq q$ . Let us assume that  $Trees(\mathcal{T}) \neq \emptyset$  and  $\Gamma \in Trees(\mathcal{T})$ . Then, the output of  $\Gamma$  for the row  $\rho$  should be equal to p and to q at the same time, but this is impossible. Therefore,  $Trees(\mathcal{T}) = \emptyset$ .

Let us assume now that the set  $\mathcal{T}$  is consistent. With  $T^{trees}(\mathcal{T})$ , we denote a decision table in which columns are labeled with attributes  $a_1, \ldots, a_n$ , and the set of rows coincides with the union of sets of rows of the tables  $T_1, \ldots, T_k$ . Each row belonging to  $T^{trees}(\mathcal{T})$  is labeled with the decision attached to this row in the tables from  $\mathcal{T}$  which this row belongs to (see Figure 2). Note that the table  $T^{trees}(\mathcal{T})$  can be constructed in polynomial time.

We now show that  $Trees(\mathcal{T}) = Trees(T^{trees}(\mathcal{T}))$ . Let  $\Gamma \in Trees(\mathcal{T})$ . Then, for any  $T_i \in \mathcal{T}$  and any row  $\rho$  belonging to  $T_i$ ,  $\Gamma$  returns the decision attached to  $\rho$  in  $T_i$ . Therefore, for any row  $\rho$  of  $T^{trees}(\mathcal{T})$ ,  $\Gamma$  returns the decision attached to  $\rho$ , i.e.,  $\Gamma \in Trees(T^{trees}(\mathcal{T}))$ . Now, let  $\Gamma \in Trees(T^{trees}(\mathcal{T}))$ . Then, for any row  $\rho$  of  $T^{trees}(\mathcal{T})$ ,  $\Gamma$  returns the decision attached to  $\rho$ . Therefore, for any table  $T_i \in \mathcal{T}$  and any row  $\rho$  of  $T_i$ ,  $\Gamma$  returns the decision attached to  $\rho$  in  $T_i$ , i.e.,  $\Gamma \in Trees(\mathcal{T})$ .

											$a_1$	$a_2$	<i>a</i> <sub>3</sub>	
	$a_1$	$a_2$	<i>a</i> <sub>3</sub>			<i>a</i> <sub>1</sub>	$a_2$	<i>a</i> <sub>3</sub>			1	1	1	1
D	1	1	1	1	D	1	1	0	2	$\tau$ trees ( $ au$ ) _	0	1	0	2
$P_1 =$	0	1	0	2	$P_2 =$	0	0	1	3	$I^{(1)}(I_1) =$	1	1	0	2
	1	1	0	2		1	0	0	3		0	0	1	3
											1	0	0	3

**Figure 2.** Joint decision table  $T^{trees}(\mathcal{T}_1)$  for the set of decision tables  $\mathcal{T}_1 = \{P_1, P_2\}$ .

## 3.3. Joint Decision Tables for Decision Rules

Let  $\mathcal{T} = \{T_1, ..., T_k\}$  be a set of decision tables, in which columns are labeled with attributes  $a_1, ..., a_n$ . A row  $\rho$  of a decision table from the set  $\mathcal{T}$  is called inconsistent if there are two tables in  $\mathcal{T}$  that contain it and if the row  $\rho$  in these tables is labeled with different decisions. Otherwise, the row  $\rho$  is called consistent.

First, we show that if the row  $\rho$  is inconsistent, then  $Rules(\mathcal{T}, \rho) = \emptyset$ . Since  $\rho$  is inconsistent, there exist two tables  $T_i$  and  $T_j$  in  $\mathcal{T}$ , such that  $\rho$  is a row of  $T_i$  labeled with a decision p,  $\rho$  is a row of  $T_j$  labeled with a decision q, and  $p \neq q$ . Let us assume that  $Rules(\mathcal{T}, \rho) \neq \emptyset$ . Then, the right-hand side of each rule from  $Rules(\mathcal{T}, \rho)$  should be equal to p and to q at the same time, but this is impossible. Therefore,  $Rules(\mathcal{T}, \rho) = \emptyset$ .

Let us assume now that the row  $\rho$  is consistent, and that it is labeled with the decision t. We denote with  $T^{rules}(\mathcal{T},\rho)$  a decision table in which columns are labeled with attributes  $a_1, \ldots, a_n$ , the first row is  $\rho$ , and the set of all other rows coincides with the union of the sets of rows of the tables  $T_1, \ldots, T_k$ , which are labeled with decisions different from t. The first row of  $T^{rules}(\mathcal{T},\rho)$  is labeled with the decision t, and all other rows are labeled with the decision t + 1 (see Figure 3). We cannot keep the initial decisions for rows that are now labeled with t + 1 since in this case, the table  $T^{rules}(\mathcal{T},\rho)$  can be inconsistent. Note that the table  $T^{rules}(\mathcal{T},\rho)$  can be constructed in polynomial time.

	$a_1$	<i>a</i> <sub>2</sub>	<i>a</i> <sub>3</sub>			$a_1$	<i>a</i> <sub>2</sub>	<i>a</i> <sub>3</sub>			<i>a</i> <sub>1</sub>	<i>a</i> <sub>2</sub>	<i>a</i> <sub>3</sub>	
$\circ$ –	1	1	1	1	0 -	1	1	1	2	$\tau$ rules ( $\tau$ a) -	1	0	0	3
$Q_1 =$	0	1	0	2	$Q_2 =$	0	1	1	3	$I^{(1)}(I_{2}, p) =$	1	1	1	4
	1	1	0	3		1	0	0	3		0	1	0	4

**Figure 3.** Joint decision table  $T^{rules}(\mathcal{T}_2, \rho)$  for the set of decision tables  $\mathcal{T}_2 = \{Q_1, Q_2\}$  and row  $\rho = (1, 0, 0)$ .

We now show that  $Rules(\mathcal{T}, \rho) = Rules(\mathcal{T}^{rules}(\mathcal{T}, \rho), \rho)$ . Let  $\rho \in Rules(\mathcal{T}, \rho)$  and  $\rho$  be equal to (3). Then, for any table  $T_i$  from  $\mathcal{T}$ , any row of  $T_i$  labeled with a decision different from t is different from  $\rho$  on at least one attribute from the set  $\{a_{i_1}, \ldots, a_{i_m}\}$ . Therefore, any row of  $T^{rules}(\mathcal{T}, \rho)$  labeled with the decision t + 1 is different from  $\rho$  on at least one attribute from the set  $\{a_{i_1}, \ldots, a_{i_m}\}$ , i.e.,  $\rho \in Rules(T^{rules}(\mathcal{T}, \rho), \rho)$ . Now, let  $\rho \in Rules(T^{rules}(\mathcal{T}, \rho), \rho)$ . Then, any row of  $T^{rules}(\mathcal{T}, \rho)$  labeled with the decision t + 1 is different from  $\rho$  on at least one attribute from the set  $\{a_{i_1}, \ldots, a_{i_m}\}$ . Therefore, for any table  $T_i$  from  $\mathcal{T}$ , any row of  $T_i$  labeled with a decision different from t is different from  $\rho$  on at least one attribute from the set  $\{a_{i_1}, \ldots, a_{i_m}\}$ . Therefore, for any table  $T_i$  from  $\mathcal{T}$ , any row of  $T_i$  labeled with a decision different from t is different from  $\rho$  on at least one attribute from the set  $\{a_{i_1}, \ldots, a_{i_m}\}$ .

## 3.4. Joint Decision Tables for Tests (Reducts)

Let  $\mathcal{T} = \{T_1, \ldots, T_k\}$  be a set of decision tables, in which columns are labeled with attributes  $a_1, \ldots, a_n$ . Each decision table from this set is consistent, but different tables from  $\mathcal{T}$  can contain equal rows labeled with different decisions. It is clear that for each table  $T_i$  from  $\mathcal{T}$ , the set of attributes  $\{a_1, \ldots, a_n\}$  is a test. Therefore,  $Tests(\mathcal{T}) \neq \emptyset$ .

We denote with  $T^{tests}(\mathcal{T})$  a decision table in which columns are labeled with attributes  $a_1, \ldots, a_n$ , the first row is filled with zeros, and the set of all other rows is constructed in the following way. For any table  $T_i$  from  $\mathcal{T}$  and any two rows  $\rho_1$  and  $\rho_2$  of  $T_i$  labeled with different decisions, we add to the table  $T^{tests}(\mathcal{T})$  the row  $c(\rho_1, \rho_2)$  filled with numbers from the set  $\{0, 1\}$ . For  $i = 1, \ldots, n$ , the row  $c(\rho_1, \rho_2)$  has the number 1 in the *i*th position if and

only if the rows  $\rho_1$  and  $\rho_2$  are different on the attribute  $a_i$ . The first row of the table  $T^{tests}(\mathcal{T})$  is labeled with the decision 1. All other rows are labeled with the decision 2 (see Figure 4). It is clear that the rows  $\rho_1$  and  $\rho_2$  are different on an attribute  $a_j$  if and only if the first row of the table  $T^{tests}(\mathcal{T})$  and the row  $c(\rho_1, \rho_2)$  are different on the attribute  $a_j$ . Note that the table  $T^{tests}(\mathcal{T})$  can be constructed in polynomial time.



**Figure 4.** Joint decision table  $T^{tests}(\mathcal{T}_3)$  for the set of decision tables  $\mathcal{T}_3 = \{R_1, R_2\}$ .

We now show that  $Tests(\mathcal{T}) = Tests(T^{tests}(\mathcal{T}))$ . Let  $B \in Tests(\mathcal{T})$ . Then, for any table  $T_i$  from  $\mathcal{T}$ , any two rows from  $T_i$  with different decisions are different on at least one attribute from B. Therefore, the first row of the table  $T^{tests}(\mathcal{T})$  is different from all other rows of the table  $T^{tests}(\mathcal{T})$  on the attributes from B, i.e.,  $B \in Tests(T^{tests}(\mathcal{T}))$ . Let  $B \in Tests(T^{tests}(\mathcal{T}))$ . Then, the first row of the table  $T^{tests}(\mathcal{T})$  is different from all other rows of the table  $T^{tests}(\mathcal{T})$  on the attributes from B. Therefore, for any table  $T_i$  from  $\mathcal{T}$ , any two rows from  $T_i$  with different decisions are different on at least one attribute from B, i.e.,  $B \in Tests(\mathcal{T})$ .

## 4. Sets of Information Systems

In this section, we deal with dispersed data represented as a finite set of information systems with equal sets of attributes.

# 4.1. Main Notions

An information system *I* is a table filled with numbers from the set  $\omega = \{0, 1, 2, ...\}$  of non-negative integers, in which columns are labeled with attributes  $a_1, ..., a_n$ . Each row  $\rho$  of the information system *I* is interpreted as an object, and the number in the intersection of the row  $\rho$  and the column  $a_i$  is interpreted as the value  $a_i(\rho)$  of the attribute  $a_i$  for the object  $\rho$ .

Any association rule over the set of attributes  $\{a_1, ..., a_n\}$  can be represented in the following form:

$$(a_{i_1} = \sigma_1) \wedge \dots \wedge (a_{i_m} = \sigma_m) \to (a_j = \sigma), \tag{4}$$

where  $a_j \in \{a_1, \ldots, a_n\}$ ,  $a_{i_1}, \ldots, a_{i_m} \in \{a_1, \ldots, a_n\} \setminus \{a_j\}$ , and  $\sigma_1, \ldots, \sigma_m, \sigma \in \omega$ . We will say that this rule is based on the attribute  $a_j$ . Rule (4) is called realizable for a row  $\rho = (\delta_1, \ldots, \delta_n) \in \omega^n$  if  $\delta_{i_1} = \sigma_1, \ldots, \delta_{i_m} = \sigma_m$ . This rule is called true for the information system *I* if for any row  $\rho'$  of *I* such that rule (4) is realizable for  $\rho', a_j(\rho') = \sigma$  (see Figure 5).

	<i>a</i> <sub>1</sub>	<i>a</i> <sub>2</sub>	<i>a</i> <sub>3</sub>	$a_4$	
	1	0	1	0	
$I_0 =$	0	0	0	1	$(a_1=0) \land (a_2=0) \to (a_4)$
	0	0	1	1	
	0	1	0	0	

**Figure 5.** Information system  $I_0$  and the association rule, which is based on the attribute  $a_4$ , true for the information system  $I_0$ , and realizable for the row (0, 0, 0, 1).

#### 4.2. Joint Information Systems for Association Rules

Let  $\mathcal{I} = \{I_1, ..., I_k\}$  be a finite nonempty set of information systems, in which columns are labeled with the same attributes  $a_1, ..., a_n$ . Let  $\rho = (\delta_1, ..., \delta_n)$  be a row of an information system from  $\mathcal{I}$  and  $a_i \in \{a_1, ..., a_n\}$ . We denote with  $Arules(\mathcal{I}, \rho, a_i)$  the set

of association rules over the set of attributes  $\{a_1, ..., a_n\}$ , each of which is based on the attribute  $a_j$ , is realizable for the row  $\rho$ , and is true for each information system from  $\mathcal{I}$ .

Our aim is to construct a so-called joint information system J, for which

$$Arules(\{J\}, \rho, a_j) = Arules(\mathcal{I}, \rho, a_j).$$
(5)

In the information system  $J = J(\mathcal{I}, \rho, a_j)$ , columns are labeled with the attributes  $a_1, \ldots, a_n$ . This information system contains row  $\rho$  and all rows  $\rho'$  from the information systems  $I_1, \ldots, I_k$ , such that  $a_j(\rho) \neq a_j(\rho')$  (we keep only one row from any group of equal rows) (see Figure 6). Note that the information system J can be constructed in polynomial time.

It is easy to show that the set of rules  $Arules(\{J\}, \rho, a_j) \cup Arules(\mathcal{I}, \rho, a_j)$  is a subset of the set *A* of rules in the following form:

$$(a_{i_1} = \delta_{i_1}) \wedge \cdots \wedge (a_{i_m} = \delta_{i_m}) \rightarrow (a_j = \delta_j),$$

where  $a_{i_1}, \ldots, a_{i_m} \in \{a_1, \ldots, a_n\} \setminus \{a_j\}$ . To show that equality (5) holds, it is enough to prove that, for any rule  $r \in A$ ,  $r \notin Arules(\{J\}, \rho, a_j)$  if and only if  $r \notin Arules(\mathcal{I}, \rho, a_j)$ . It is clear that each rule from A is based on the attribute  $a_j$  and is realizable for the row  $\rho$ . Let  $r \notin Arules(\{J\}, \rho, a_j)$ . Then, the rule r is not true for J, and there exists a row  $\rho'$  from J such that r is realizable for  $\rho'$  and  $a_j(\rho) \neq a_j(\rho')$ . It is clear that  $\rho'$  is a row from an information system  $I_i$  from  $\mathcal{I}$ . Then, r is not true for  $I_i$  and  $r \notin Arules(\mathcal{I}, \rho, a_j)$ . Let  $r \notin Arules(\mathcal{I}, \rho, a_j)$ . Then, there exists an information system  $I_i \in \mathcal{I}$  for which r is not true, and there exists a row  $\rho'$  from  $I_i$  such that r is realizable for  $\rho'$  and  $a_j(\rho) \neq a_j(\rho')$ . It is clear that  $\rho'$  is a row from the information system J. Then, r is not true for J, and  $r \notin Arules(\{J\}, \rho, a_j)$ . Thus, the equality (5) holds.



**Figure 6.** Joint information system  $J(\mathcal{I}, \rho, a_3)$  for the set of information systems  $\mathcal{I} = \{I_1, I_2\}$ , row  $\rho = (1, 0, 0)$ , and attribute  $a_3$ .

## 5. Conclusions

In this simple methodological paper, we have shown the problem of studying common decision trees for a dispersed set of decision tables with equal sets of attributes and how to reduce this to the study of decision trees for a single decision table. We accomplished the same for common decision rules and tests (reducts). The proposed approach allows us to generalize known methods in the study of single decision tables to the case of dispersed tables with equal sets of attributes.

We also showed the problem of studying common association rules for a dispersed set of information systems with equal sets of attributes and how to reduce this to the study of association rules for a single information system. The proposed approach allows us to generalize known methods in the study of association rules for single information systems to the case of dispersed information systems with equal sets of attributes.

The presented idea is different from the methods offered in the framework of distributed data mining or data warehouses. In our approach, the cost of communication in a distributed environment is limited to the construction of a joint tabular form. Then, depending on the aim of the data analysis, different existing algorithms for the induction of decision trees, rules, reducts, or association rules can be used. In the case of data warehouses, the main application is the use of OLAP tools for supporting business decisions. In the case of distributed data mining, collaboration among agents in the entire system and learning at the local level are important factors that are omitted in the proposed approach. Future research will be connected with developing an algorithm for the induction of decision rules from distributed data. The proposed idea will be different from the one presented in this paper, since decision rules will be induced from a set of decision tables without the process of transforming the distributed data into a joint tabular form.

**Author Contributions:** Conceptualization, all authors; methodology, all authors; formal analysis, all authors; investigation, all authors; writing—original draft preparation, M.M. and B.Z.; supervision, M.M.; project administration, M.M.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: Research funded by King Abdullah University of Science and Technology.

Data Availability Statement: Not applicable.

Acknowledgments: The research work reported in this publication was supported by King Abdullah University of Science and Technology (KAUST).

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

# References

- 1. Fu, Y. Distributed data mining: An overview. Newsl. IEEE Tech. Comm. Distrib. Process. 2001, 4, 5–9.
- Moshkov, M. Decision trees and reducts for distributed decision tables. In *Monitoring, Security, and Rescue Techniques in Multiagent* Systems, MSRAS 2004, Plock, Poland, 7–9 June 2004; Advances in Soft Computing; Dunin-Keplicz, B., Jankowski, A., Skowron, A., Szczuka, M.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 28, pp. 239–248.
- Ślęzak, D. Decision value oriented decomposition of data tables. In *Foundations of Intelligent Systems, 10th International Symposium, ISMIS '97, Charlotte, NC, USA, 15–18 October 1997, Proceedings;* Lecture Notes in Computer Science; Ras, Z.W., Skowron, A., Eds.; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1325, pp. 487–496.
- 4. Pawlak, Z. *Rough Sets-Theoretical Aspects of Reasoning about Data;* Theory and Decision Library: Series D; Kluwer: Alphen aan den Rijn, The Netherlands, 1991; Volume 9.
- 5. Pawlak, Z.; Skowron, A. Rudiments of rough sets. Inf. Sci. 2007, 177, 3–27. [CrossRef]
- Moshkov, M.; Zielosko, B. Combinatorial Machine Learning—A Rough Set Approach; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2011; Volume 360.
- 7. AbouEisha, H.; Amin, T.; Chikalov, I.; Hussain, S.; Moshkov, M. *Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining*; Intelligent Systems Reference Library; Springer: Berlin/Heidelberg, Germany, 2019; Volume 146.
- Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. Classification and Regression Trees; Chapman and Hall/CRC: Boca Raton, FL, USA, 1984.
- 9. Moshkov, M. Time complexity of decision trees. In *Trans. Rough Sets III*; Lecture Notes in Computer Science; Peters, J.F., Skowron, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3400, pp. 244–459.
- 10. Rokach, L.; Maimon, O. *Data Mining with Decision Trees-Theory and Applications*; Series in Machine Perception and Artificial Intelligence; World Scientific: Singapore, 2007; Volume 69.
- 11. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules in large databases. In *VLDB*; Bocca, J.B., Jarke, M., Zaniolo, C., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1994; pp. 487–499.
- 12. Alsolami, F.; Amin, T.; Moshkov, M.; Zielosko, B.; Žabiński, K. Comparison of heuristics for optimization of association rules. *Fundam. Inform.* **2019**, *166*, 1–14. [CrossRef]
- Moshkov, M.; Piliszczuk, M.; Zielosko, B. Greedy algorithm for construction of partial association rules. *Fundam. Informaticae* 2009, 92, 259–277. [CrossRef]
- Nguyen, H.S.; Ślęzak, D. Approximate reducts and association rules-correspondence and complexity results. In *RSFDGrC*; Lecture Notes in Computer Science; Zhong, N., Skowron, A., Ohsuga, S., Eds.; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1711, pp. 137–145.
- 15. Wieczorek, A.; Słowiński, R. Generating a set of association and decision rules with statistically representative support and anti-support. *Inf. Sci.* 2014, 277, 56–70. [CrossRef]
- 16. Zielosko, B. Application of dynamic programming approach to optimization of association rules relative to coverage and length. *Fundam. Inform.* **2016**, *148*, 87–105. [CrossRef]
- Moshkov, M. Common decision trees, rules, and tests (reducts) for dispersed decision tables (to appear). In Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022), Verona, Italy, 7–9 September 2022.
- Moshkov, M.; Zielosko, B.; Tetteh, E.T. Common association rules for dispersed information systems (to appear). In Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022), Verona, Italy, 7–9 September 2022.

- 19. Amuthabala, P.; Santhosh, R. Robust analysis and optimization of a novel efficient quality assurance model in data warehousing. *Comput. Electr. Eng.* **2019**, *74*, 233–244. [CrossRef]
- 20. Theodorou, V.; Jovanovic, P.; Abellò, A.; Nakuçi, E. Data generator for evaluating ETL process quality. *Inf. Syst.* 2017, 63, 80–100. [CrossRef]
- Cuzzocrea, A. Editorial: Models and algorithms for high-performance distributed data mining. J. Parallel Distrib. Comput. 2013, 73, 281–283. [CrossRef]
- 22. Lin, K.W.; Chung, S.H. A fast and resource efficient mining algorithm for discovering frequent patterns in distributed computing environments. *Future Gener. Comput. Syst.* 2015, *52*, 49–58. [CrossRef]
- Kargupta, H.; Kamath, C.; Chan, P. Distributed and parallel data mining: Emergence, growth, and future directions. In Advances in Distributed and Parallel Knowledge Discovery; AAAI/MIT Press: Cambridge, MA, USA, 2000; pp. 409–416.
- 24. Urmela, S.; Nandhini, M. A framework for distributed data mining heterogeneous classifier. *Comput. Commun.* 2019, 147, 58–75. [CrossRef]
- 25. Vilalta, R.; Giraud-Carrier, C.; Brazdil, P. Meta-learning-concepts and techniques. In *Data Mining and Knowledge Discovery* Handbook; Springer: Boston, MA, USA, 2010; pp. 717–731.
- Chikalov, I.; Lozin, V.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; Intelligent Systems Reference Library; Springer: Berlin/Heidelberg, Germany, 2013; Volume 41.
- 27. Stefanowski, J.; Vanderpooten, D. Induction of decision rules in classification and discovery-oriented perspectives. *Int. J. Intell. Syst.* **2001**, *16*, 13–27. [CrossRef]
- 28. Pawlak, Z.; Skowron, A. Rough sets and Boolean reasoning. *Inf. Sci.* 2007, 177, 41–73. [CrossRef]
- 29. Han, J.; Kamber, M. Data Mining: Concepts and Techniques; Morgan Kaufmann: San Francisco, CA, USA, 2000.
- 30. Żabiński, K.; Zielosko, B. Decision rules construction: Algorithm based on EAV model. Entropy 2021, 23, 14. [CrossRef] [PubMed]
- 31. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* 2013, 39, 261–283. [CrossRef]
- 32. Quinlan, J.R. C4.5: Programs for Machine Learning; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993.
- 33. Rivera-Lopez, R.; Canul-Reich, J.; Mezura-Montes, E.; Cruz-Chávez, M.A. Induction of decision trees as classification models through metaheuristics. *Swarm Evol. Comput.* **2022**, *69*, 101006. [CrossRef]
- Guyon, I.; Gunn, S.; Nikravesh, M.; Zadeh, L. (Eds.) Feature Extraction: Foundations and Applications; Studies in Fuzziness and Soft Computing; Springer: Berlin/Heidelberg, Germany, 2006; Volume 207.
- 35. Liu, H.; Motoda, H. Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series); Chapman & Hall/CRC: Boca Raton, FL, USA, 2007.
- Stańczyk, U.; Zielosko, B.; Żabiński, K. Application of greedy heuristics for feature characterisation and selection: A case study in stylometric domain. In Proceedings of the Rough Sets-International Joint Conference, IJCRS 2018, Quy Nhon, Vietnam, 20–24 August 2018; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11103, pp. 350–362.
- 37. Jia, X.; Shang, L.; Zhou, B.; Yao, Y. Generalized attribute reduct in rough set theory. *Knowl.-Based Syst.* 2016, 91, 204–218. [CrossRef]
- 38. Wróblewski, J. Theoretical foundations of order-based genetic algorithms. Fundam. Inform. 1996, 28, 423–430. [CrossRef]
- 39. Zielosko, B.; Piliszczuk, M. Greedy algorithm for attribute reduction. Fundam. Inform. 2008, 85, 549–561.
- Grzegorowski, M.; Slęzak, D. On resilient feature selection: Computational foundations of r-C-reducts. Inf. Sci. 2019, 499, 25 44. [CrossRef]
- 41. Lee, A.J.T.; Hong, R.W.; Ko, W.M.; Tsao, W.K.; Lin, H.H. Mining spatial association rules in image databases. *Inf. Sci.* 2007, 177, 1593–1608. [CrossRef]
- 42. Han, J.; Fu, Y. Discovery of multiple-level association rules from large databases. In *VLDB*; Dayal, U., Gray, P.M.D., Nishio, S., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1995; pp. 420–431.
- Agrawal, R.; Imieliński, T.; Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 25–28 May 1993; pp. 207–216.
- 44. Han, J.; Pei, J.; Yin, Y.; Mao, R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.* **2004**, *8*, 53–87. [CrossRef]
- Borgelt, C. Simple algorithms for frequent item set mining. In *Advances in Machine Learning II*; Studies in Computational Intelligence; Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 263, pp. 351–369.
- 46. Herawan, T.; Deris, M.M. A soft set approach for association rules mining. *Knowl.-Based Syst.* 2011, 24, 186–195. [CrossRef]
- 47. Mattiev, J.; Kavsek, B. Coverage-based classification using association rule mining. Appl. Sci. 2020, 10, 7013. [CrossRef]