TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology

Fjodor Ševtšenko 144334

ZERO FACTORS FREE RULES ALGORITHM: THE STUDY OF CLASSIFICATION FUNCTION

Master's thesis

Supervisor: Grete Lind
MSc
Rein Kuusik
PhD

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Fjodor Ševtšenko 144334

NULLFAKTORIVABADE REEGLITE ALGORITM: KLASSIFITSEERIMISVÕIME UURING

Magistritöö

Juhendaja: Grete Lind
MSc
Rein Kuusik
PhD

Tallinn 2017

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Fjodor Ševtšenko

01.05.2017

Abstract

The need of classifying the unknown set of objects is a very important skill of a modern information system with integrated artificial intelligence functionality. One system should be able to help a medical specialist to make a diagnosis, other system should be ready to identify the creditability of a person applying for a loan, third system should be good in finding out the fraud attacks among the list of requests sent.

The classification of unknown data set could be done using a rule-based machine learning method. ZFFR algorithm is one of such methods' representative. Method is directed to solve a data analysis task to describe a data set under analysis: who are they? What distinguish them (data objects under analysis)? There is no methodology for ZFFR algorithm on the field of how to classify the unknown data set based on rules. The absence of the methodology means, that there is no information about how accurately received rules are able to perform the recognition's function.

The main goal of the thesis was to develop a classification method acting on the rules received by ZFFR algorithm, estimate the developed method's accuracy and rank it compared to proposed benchmark table. Author set the hypothesis: more rules are generated, more accurately the worked out method is able to recognize the unknown objects. The testing of the hypothesis required making the improvements in ZFFR algorithm's time-effective characteristics in order to be able to generate maximum number of rules on the lowest levels of thresholds.

This thesis is written in English and is 105 pages long, including 5 chapters, 32 figures and 15 tables.

Annotatsioon

Nullfaktorivabade reeglite algoritm: klassifitseerimisvõime uuring

Üks olulisematest funktsioonidest tehisintellekti omadustega infosüsteemidel on tundmatu objektide hulka klassifitseerimine. Ühed süsteemid aitavad meditsiinitöötajatel diagnoosi panna, teised annavad hinnangu laenutaotleja krediidivõimele, kolmandad suudavad tuvastada pettuse rünnakuid.

Tundmatuid objekte saab klassifitseerida, kasutades masinõppe meetodeid. Nimetatud meetodite esindajate hulka kuulub ka ZFFR algoritm. Selle meetodi eesmärk on objektide kirjeldamine vastates järgmistele küsimustele. Kes nad on? Millised on nende objektide põhilised omadused? Mille poolest need objektid üksteisest erinevad? Hetkel ei ole olemas metodoloogiat, kuidas tundmatuid objekte klassifitseerida, kasutades ZFFR algoritmi poolt genereeritud reegleid. See tähendab, et puudub ka informatsioon, kui täpselt saadud reeglid suudavad täita klassifitseerimisfunktsiooni.

Magistritöö põhiline eesmärk on välja arendada ZFFR algoritmiga leitud reeglitel põhinev klassifitseerimismeetod, hinnata meetodi täpsust ning võrrelda saadud täpsust teiste klassifitseerimismeetodite tulemustega.

Autor püstitab hüpoteesi: mida rohkem reegleid saadakse, seda täpsem klassifitseerimistulemus saavutatakse. Hüpoteesi testimine nõuab ZFFR algoritmi reeglite genereerimise kiiruse parandamist, kuna reeglite genereerimisparameeter – sageduspiir (reegli esinemissagedus) – peab olema võimalikult madal.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 105 leheküljel, 5 peatükki, 32 joonist, 15 tabelit.

List of abbreviations and terms

ZFFR	Zero Factors Free Rules
TL	Total number of letters observed
MA1	Number of letters correctly classified under condition 1
AC1	Accuracy rate calculated under condition 1
MA2	Number of letters correctly classified under condition 2
AC2	Accuracy rate calculated under condition 2
MA3	Number of letters correctly classified under condition 3
AC3	Accuracy rate calculated under condition 3
R	Number of rules found
RT	Time spent for rules' generation
CT	Time spent for classifying the test-objects

Table of contents

List of figures

List of tables

1 Introduction

The need of classifying the unknown set of objects is a very important "skill" of a modern information system with integrated artificial intelligence functionality. One system should be able to help a medical specialist to make a diagnosis, other system should be ready to identify the creditability of a person applying for a loan, third system should be good in finding out the fraud attacks among the list of requests sent.

The classification of unknown data set could be done using a rule-based machine learning method. ZFFR algorithm is one of such methods' representative. Method is directed to solve a data analysis task to describe a data set under analysis: who are they? What distinguish them (data objects under analysis)? The algorithm's nature was described by Grete Lind and Rein Kuusik in "Algorithm for Finding Zero Factor Free Rules" [1]. Based on tabular formatted data set ZFFR algorithm is able to generate associative rules in an "IF-THEN" form, dividing the factors into the factor, excluded-factor and zero-factor categories. The resulted rule based on ZFFR algorithm could have a form of "IF Factor-Condition THEN Class AND Zero-Factor-Condition AND NOT Excluded-Factor-Condition". The time-effective characteristics in rules' generation were investigated by Liisa Jõgiste in "Prototyping of Zero-factor based DA" [2].

Problem. There is no methodology for ZFFR algorithm on the field of how to classify the unknown data set based on rules. The absence of the methodology means, that there is no information about how accurately received rules are able to perform the recognition's function.

Targets. Work out the method abled to classify the set of unknown objects based on rules received by ZFFR algorithm. Estimate the method's accuracy.

Hypothesis. The classification's method should use the rules, generated by ZFFR algorithm. More rules are generated, more accurately the worked out method is able to recognize the unknown objects.

To check the hypothesis, it will be necessary to generate rules, using the very low levels of thresholds. This will require to perform a solid amount of computations, especially for deeply structured data sets. In order to implement this task in a reasonable time frame, the time-efficiency characteristics of ZFFR algorithm should be improved. This can be considered as a *sub-target*.

Data set used. Benchmark. How accurately ZFFR algorithm is able to classify the set of unknown objects? To answer this question, it is important to find a such data set, that was already classified by some other algorithms. In this case the necessary benchmark will be obtained, which could help to estimate the quality of the classification function of ZFFR method.

One of the good examples of such data set could be "letter recognition" data set [3]. This data was described and used by Peter W. Frey and David. J. Slate in "Letter Recognition Using Holland-style Adaptive Classifiers" in 1991 [4]. It consists of 20000 rows where each row describes one particular English alphabet's letter by 16 primitive numerical attributes, "scaled to fit into a range of integer values from 0 through 15". The scientists used 16000 items for model [HSAC] training and 4000 items for testing. The best accuracy result for this data set was 82,7%.

In year 2009 Chunlin Liang, Lingxi Peng, Yindie Hong and Jing Wang from Software School, Guangdong Ocean University (China) classified the same data set using algorithm based on artificial immune, called LEBAI [5]. The authors from Software School in their work described not only the used algorithm's characteristics but also provided us a benchmark table for the accuracy results obtained by other algorithms for "letter recognition" data set [3]. This information together with references to sources is provided in Table 1.

Algorithm	**Accuracy (%)**
HSAC [4]	82,70
Genetic programming [6]	92,00
Neural networks [7]	94,13
FNNC [8]	95,67
LEBAI [5]	95,87

Table 1. The comparison of letters recognition accuracy [5]

The best result of accuracy of 95,87%, as was mentioned, was achieved by using the artificial immune algorithm. The principles of this algorithm were taken from the biological organisms. The biology in general is a good source for finding out a well-performed templates for processing the information. Two representatives of such kind of approach are seen in Table 1 as well.

The first one is neural networks that could achieve the level of accuracy of 94,13%. Neural network is a mathematical model that interprets a work of neural network located in the brain. The model uses such terms from biology like neurons, axons, synapses and presents a very powerful tool for example in area of making predictions. At the same time, the developing of neural networks is not a trivial task. The training of a model could take quite a long time. Neural network is acting as a "black box" that means that the reasons of existence of associations of data elements could not be easily described. Another example of a good prediction's performance (92,0%) is a genetic programming. This is the second representative of approach the principles of which were taken from the biology. Genetic algorithms are based on the theory of evolutions and operates such terms as DNA, genes, protein, chromosomes, mutations. This approach uses the optimization's techniques for finding the best solution from the group of the alternatives.

One of the Master's thesis' objectives is to rank the classification function of ZFFR based on the given benchmark table.

Author has divided the work into three chapters. The first chapter describes the methodology used. In this part author explains the nature of main data structures used in ZFFR algorithm's implementation. Then here are described new methods and principles of achieving the time-efficient characteristics of algorithm, as well as multi-threaded technique that could provide a significant time reduction in rules generation. In the last part of the first chapter author describes the created method of classification of unknown objects together with accuracy calculation technique.

The second chapter contains the information about the results achieved. Here could be found a numerical and a graphical presentations of outcomes regarding the set hypothesis and targets placed in thesis.

The third chapter contains the discussions about the main achievements received in a field of classification and ZFFR algorithm's time efficient characteristics in rules' generation. Here will be explained the nature of rules conflict problem together with a possible solution to the named problem.

The last chapter contains the conclusions' part.

2 Methodology

The main target of the thesis is to work out the method, abled to classify the set of unknown objects based on rules, received by ZFFR algorithm and estimate this method's accuracy. The needed to check hypothesis states that more rules are generated – more accuracy of classification could be achieved. Set hypothesis requires from ZFFR algorithm to generate rules, using the lowest levels of thresholds, preferably 1.

Single-threaded implementation of ZFFR algorithm like in [2], especially for highly structured data sets like for "letter recognition" data set [3], could lead to "out of memory" problem. That is why, before presenting the classification's method, it is necessary to upgrade the time-efficient characteristics of ZFFR algorithm.

The methodological chapter consists of three parts. The first part describes the main data structures used in author's implementation of ZFFR algorithm. The second part presents new methods and principles used to increase the time-efficient characteristics of the algorithm together with multi-threaded technique. The proposed classification's method together with accuracy calculation's logic will be described in the last part of this chapter.

2.1 The mapping between the data sets and the data structures

ZFFR algorithm processes a data set that has a tabular format. To guarantee the efficient work of rules' generation, it is very important to "explain" to the algorithm the meanings of every necessary field of an initial table. In other words, a data set should be interpreted as a data structure. The first part of this subchapter describes the data structure used for the Factor as a key element of the algorithm's logic.

The result of ZFFR algorithm's execution is a rules' base. In order to be able to make queries to a rules' base or, in particular, to implement the classification's method, a rule's data set should be transformed into a suitable data structure as well. The detailed

information about what data structure is used for saving a rule is presented in the second part of this subchapter.

In order to keep the description of the methodology simple, author provides the explanation of every part with an example. The examples are done based on well-known Quinlan's data set [9].

2.1.1 The data structure for the interpretation of Factor

The data from the Quinlan's data set is presented in Table 2. It has a tabular format, where each column has its own name and meaning.

id	**oid**	**Outlook (Ou)**	**Temperature (Te)**	**Humidity (Hu)**	**Windy (Wi)**	**Class (Cl)**
1	1	sunny	hot	high	FALSE	N
2	2	sunny	hot	high	TRUE	N
3	3	overcast	hot	high	FALSE	P
4	4	rain	mild	high	FALSE	P
5	5	rain	cool	normal	FALSE	P
6	6	rain	cool	normal	TRUE	N
7	7	overcast	cool	normal	TRUE	P
8	8	sunny	mild	high	FALSE	N
9	9	sunny	cool	normal	FALSE	P
10	10	rain	mild	normal	FALSE	P
11	11	sunny	mild	normal	TRUE	P
12	12	overcast	mild	high	TRUE	P
13	13	overcast	hot	normal	FALSE	P
14	14	rain	mild	high	TRUE	N

Table 2. Quinlan's data set [9]

The row identification number is marked as "id". "Oid" is an object's identification number. "Outlook (Ou)", "Temperature (Te)", "Humidity (Hu)" and "Windy (Wi)" are the names of used *attributes*. The last column called "Class (Cl)" defines a *class*. The intersection cell between "oid" and attribute contains a *value*. The intersection cell between "oid" and "Class (Cl)" is a *value of class*.

Before transforming the initial table into the data structure, the attributes' names and the name of the class should be interpreted as numbers. For example, the attribute's name "Outlook (Ou)" is mapped with number 1. The attribute's name "Temperature (Te)" is mapped with number 2, "Humidity (Hu)" - with number 3 and so on. The same technique should be used for the values, even for those who already have the numeric nature. For example, value "sunny" for "Outlook (Ou)" and "oid" 1 is mapped with 1, value "overcast" for "Outlook (Ou)"" and "oid" 3 is mapped with 2, value "rain" for "Outlook (Ou)"" and "oid" 4 is mapped with 3. Doing the mapping for all values (text and numeric) simplifies the back-transforming procedure. It is done for all elements without need to remember what value should be replaced, what value should not. Such mapping is able to increase the algorithm's time-efficiency characteristics. When the rules are generated, the opposite transformation takes place.

The key element of author's ZFFR algorithm's implementation is a data structure called Factor. The Factor includes the data positions for the required fields from the initial data set. Also the Factor contains the slots for the additional information what is set during the Factor's processing activities done by ZFFR algorithm. The Factor consists of 10 elements (Figure 1).

```
01: CLASS Factor
02:   id: String
03:   idset: Set[String]
04:   attributeId: Int
05:   value: Int
06:   frequency: Int
07:   objects: Set[Int]
08:   parents: Map[Int, Int]
09:   class: Map[Int, Int]
10:   zeros: Map[Int, Int]
11:   excludes: Map[Int, Int]
```

Figure 1. Factor's data structure

In order to understand how Factor's elements are set, it is essential to remember ZFFR algorithm's logic [1] [2], which requires the calculation of frequency's table. The example presented in Table 3 is aimed to simplify the understanding of the Factor's nature. The content of Table 3 includes the first seven objects of Quinlan's data set together with the Factor's identification process. The attributes', class's names and their values are replaced with numbers.

steps	**oid**	**1**	**2**	**3**	**4**	**5**
0	1	1	3	1	2	2
	2	1	3	1	1	2
	3	2	3	1	2	1
	4	3	2	1	2	1
	5	3	1	2	2	1
	6	3	1	2	1	2
	7	2	1	2	1	1
1	**1**	**2**	**3**	**4**	**3**	**4**
	2	**2**	**1**	**3**	**4**	**3**
	3	**(3)**	**3**	**0**	**0**	**0**
2	3	1				
	4	3	2	1	2	1
	5	3	1	2	2	1
	6	3	1	2	1	2
3	**1**	**0**	**(2)**	**1**	**0**	**2**
	2	**0**	**1**	**2**	**2**	**1**
	3	**3**	**0**	**0**	**1**	**0**
	2	1	1			
4	5	3	1	2	2	1
	6	3	1	2	1	2
5	**1**	**0**	**2**	**0**	**1**	**1**
	2	**0**	**0**	**2**	**1**	**1**
	3	**2**	**0**	**0**	**0**	**0**

Table 3. The Factor's identification process (for the first seven objects)

The initial data set is given on the Step 0. The Step 1 calculates the frequencies' table. The leading Factor is marked green. The "attributeId" of this factor equals 1, the

“value” equals 3, the “frequency” equals 3, the “objects” set equals [4,5,6] (Step 2). There are no parent elements for this factor, “parents” equals empty map [,]. “Class”, “zeros” and “excludes” are also empty maps [,]. The factor’s “id” is a combination from “attributeId” and “value” separated with a column, which equals “1:3”. The factor’s “idset” contains the “id” element [“1:3”]. The factor class with initialized values is presented in Figure 2.

```
01: CLASS Factor
02:   id: “1:3”
03:   idset: [“1:3”]
04:   attributeId: 1
05:   value: 3
06:   frequency: 3
07:   objects: [4,5,6]
08:   parents: [,]
09:   class: [,]
10:   zeros: [,]
11:   excludes: [,]
```

Figure 2. Factor's example A, without parents

Suppose the threshold is 2. Class is not defined, then the next leading factor is 2 (marked green, Step 3). On steps 4 and 5 the new factor is defined. The “attributeId” of this factor equals 2, the “value” equals 1, the “frequency” equals 2, the “objects” set equals [5,6]. “Class”, “zeros” and “excludes” are empty maps [,]. For this factor the parent element is defined. It is a tuple of the “attributeId” and “value” of the previously defined factor – (1,3). It means that the “parents” map equals [(1,3)]. The “idset” now contains two elements [“1:3”,”2:1”]. The “id” is received as an alphabetically sorted combination of “idset” and equals “1:3 2:1“. The factor class with initialized values is presented in Figure 3.

```
01: CLASS Factor
02:   id: “1:3 2:1”
03:   idset: [“1:3”,”2:1”]
04:   attributeId: 2
05:   value: 1
06:   frequency: 2
07:   objects: [5,6]
08:   parents: [(1,3)]
09:   class: [,]
10:   zeros: [,]
11:   excludes: [,]
```

Figure 3. Factor's example B, with parents

The Factor’s identification process is very important. It provides the opportunities to decrease the amount of work, doing by algorithm. In other words, the algorithm “knows” exactly, which Factor is processed (work is already done) and which Factor waits for processing (work is already assigned).

2.1.2 The data structure for the interpretation of Rule

One of the most interesting features of ZFFR algorithm is getting the rule as a logical construction of factors, zero factors, excluded factors and classes. The example of such rule is presented in Figure 4.

```
“class”:”P” &
“humidity”:”normal” &
NOT “outlook”:”overcast” <= “temperature”:”cool” & ”windy”:”FALSE”
```

Figure 4. Rule's static presentation

The class of the rule is marked with a colour **red**, zero factor is marked **green**, excluded factor is marked **blue** and class defined factors are marked **grey**. This static representation of a rule suites well for human, but not for algorithm. If there is a need to process the rules’ base presented in a such way, it is necessary to create a method, which could help the algorithm to identify the nature of included elements. Usually, the count of rules received could be more than half of million. It means that method should be optimized for making complicated queries in order to meet the time-efficient requirements.

One of the alternatives to the creation of processing method could be adapting the way of rule's presenting to already existing queries-making systems, for example PostgreSQL database.

When the "class" element of the Factor is found, the Rule can be created. The example of the Rule's data structure together with values is presented in Figure 5.

```
01: CLASS Rule
02:   id:         7859692
03:   class_j:    [{"class":"P"}]
04:   factors_j:  [{"temperature":"cool"}, {"windy":"FALSE"}]
05:   zeros_j:    [{"humidity":"normal"}]
06:   excludes_j: [{"outlook":"overcast"}]
07:   oids_j:     ["9","5"]
```

Figure 5. Rule's data structure

The Rule's data structure is actually a data structure over a data structure. Every Rule's element (except "id") is saved using JSON [10] representation of data. Figure 5 contains a real example of the rule generated based on Quinlan's data set [9]. Using JSON format is very practical, especially when it is required to make queries on received rules' base or use rule's base for classification purposes. Such structure can be easily saved in PostgreSQL database. The analyses of the rules can be done using the standard SQL techniques [11].

2.2 The principles and techniques for improving the time-efficiency characteristics of ZFFR algorithm

In order to check the hypothesis about the influence of number of rules on the figure of accuracy, it is essential to have time-efficient ZFFR algorithm that is able to generate rules on very low levels of threshold. The time-efficient characteristics could be achieved by following two principles: "don't perform work that is already done" and "don't assign work that is already assigned". The next subchapters are about the techniques that guarantee the implementation of the named principles, starting from the general description of the structure of ZFFR algorithm and rules' optimization technique. The possible implementation of parallel techniques for the algorithm will be described in the last subchapter.

2.2.1 ZFFR algorithm's structure

Comparing with ZFFR algorithm's structure used in [1], author's implementations doesn't contain the recursions. In Figure 6 is presented the method "generateRules" as an entry point of algorithm.

```
01: stack: Stack[Factor]

02: toVisitFactors: Set[String]
03: visitedFactors: Set[String]

04: describedObjects: Map[Int, Set]
05: describedFactors: Map[String, Set]

06: ruleFactors: Map[Int, Set]
07: ruleObjects: Map[Int, Set]

08: rules: Map[Int, Factor]

09: FUNCTION generateRules()
10:   factors = getNoZeroNoExclFactorsData()
11:   FOR factor IN factors
12:     stack.push(factor)
13:   NEXT
14:   WHILE stack NOT EMPTY
15:     topFactor = stack.pop()
16:     IF (isProbableClassInside(topFactor))
17:       checkForClass(topFactor)
18:       setFactorVisited(topFactor)
19:     END IF
20:   LOOP
21: END FUNCTION
```

Figure 6. ZFFR algorithm: generateRules method

The recursion here is replaced by stack-while structure. Such approach allows to regulate the "depth" of data structure. User can rewrite the implementation of the stack by providing the additional functionality of defining the count of contained elements. If there is a need to add an element to a "full-stack", then new stack's instance can be created and required element is added to the new one. In this case instead of one stack there is a list of stacks. This approach is aimed to control the amount of memory allocated to one variable. If the algorithm is implemented in programming language working on Java Virtual Machine (JVM), then additionally could be set a value of JVM options, regulating the available size of heap. Such activity could prevent from "out of memory" errors.

First of all, the method gets “factors” that responds to the required threshold (line 10). These “factors” are added to the “stack” (line 12). In terms of mentioned principles this means the assignment of the work to do. While “stack” is not empty (line 14), the method takes the first available factor (line 15) form the “stack” and checks the class’s availability for the current “topFactor” (“checkForClass”). Author’s realization of ZFFR algorithm assumes the generating of rules only for factors that have the class definition behind. After the completion of “checkForClass” call, “topFactor” is set “visited” (line 18). In terms of mentioned principles this means the work for this factor is marked done. The checking of condition “isProbableClassInside” is skipped for now. The detailed explanation will be provided in the last subchapter that explains a technique, regarding the parallel implementation of the algorithm.

The special attention should be drawn to “checkForClass” method (Figure 7).

```
01: FUNCTION checkForClass(topFactor)
02:   classFactor = getClassData(topFactor)
03:   IF classFactor NOT EMPTY
04:     ruleId += 1
05:     checkForZeroFactor(topFactor)
06:     checkForExclFactor(topFactor)
07:     optimizeRulesBase(classFactor)
08:     setObjectsDescribed(classFactor, ruleId)
09:     setFactorsDescribed(classFactor, ruleId)
10:     ruleFactors += (ruleId, Set(classFactor.idset))
11:     ruleObjects += (ruleId, Set(classFactor.objects))
12:     rules += (ruleId, classFactor)
13:   ELSE
14:     factors = getNoZeroNoExclFactorsData(topFactor)
15:     FOR factor IN factors
16:       IF isProbableClassInside(factor)
17:         IF NOT isFactorToVisit(factor)
18:           setFactorToVisit(factor)
19:           IF NOT areObjectsDescribed(factor)
20:             stack.push(factor)
21:           END IF
22:         END IF
23:       END IF
24:     NEXT
25:   END IF
26: END FUNCTION
```

Figure 7. ZFFR algorithm: checkForClass method

At first, “getClassData” method is called (line 02). If class is found, then “ruleId” is defined (line 04). Line 05 and line 06 check, if there are zero-factors or excluded factors presented for this rule. The optimization of the rules’ base is done in line 07. Rules’ optimization will be separately described in subchapter 2.2.2. Line 08 and line 09 are responsible for marking objects and factors with current “ruleId”. Line 10 sets new ruleId-factors combination. Line 11 sets new ruleId-objects combination. Line 12 adds new rule to rules’ base.

If class is not found, then children factors of the “topFactor” are received (line 14). Every child is checked with method “isProbableClassInside” (subchapter 2.2.5) and with method “isFactorToVisit” (subchapter 2.2.4) in lines 16 and 17 respectively. In line 18 factor is set to visit, what means new work is assigned. Before adding a new factor to the stack (line 20), line 19 calls the method “areObjectsDescribed” (subchapter 2.2.3).

2.2.2 The rules’ optimization technique

ZFFR algorithm described in [1] does not contain rules’ compression. The compression activity is done as separate flow. Author’s implementation of the algorithm includes the rule’s compression as a built-in technique. From the one hand, the integration of the compression helps to receive already optimized rules’ base. From the other hand, it helps to perform less work, maintaining the state of “describedObjects” (subchapter 2.2.3).

Suppose, that there is some Map with rules (Figure 8). These rules are received using Quinlan’s data set [9].

```
01: (1, {Ou:2})
02: (2, {Te:3, Ou:2})
03: (3, {Te:3, Ou:1})
04: (4, {Ou:1, Hu:2})
05: (5, {Ou:1, Hu:1})
06: (6, {Ou:3, Wi:1})
07: (7, {Ou:3, Te:2, Wi:2})   ~~(7, {Ou:3, Te:2, Wi:2})~~
08: (8, {Ou:3, Hu:2, Wi:2})   ~~(8, {Ou:3, Hu:2, Wi:2})~~
09: (9, {Ou:3, Wi:2})
```

Figure 8. ZFFR algorithm's instance variable "ruleFactors": Map[Int, Set]

The rules marked black describe the initial state of the Map. The lines marked red describe the changes inside the Map. The first element of the map is “ruleId”, the second element of the map is set of “factorId” defining the particular rule.

In Figure 9 there is a map of “factorId” and set of “ruleId”. The lines marked black describe the initial state of the Map. The lines marked red describe the changes inside the Map.

```
01: (Ou:2, {1})
02: (Te:1, {2})
03: (Wi:2, {2,7,8})  ~~(Wi:2, {2,7,8})~~   (Wi:2, {2,9})
04: (Te:3, {3})
05: (Ou:1, {3,4,5})
06: (Hu:2, {4,8})    ~~(Hu:2, {4,8})~~     (Hu:2, {4})
07: (Hu:1, {5})
08: (Ou:3, {6,7,8})  ~~(Ou:3, {6,7,8})~~   (Ou:3, {6,9})
09: (Wi:1, {6})
10: (Te:2, {7})      ~~(Te:2, {7})~~       (Te:2, {})
```

Figure 9. ZFFR algorithm's instance variable "describedFactors": Map[String, Set]

Suppose, that a new rule is received. The rule’s id is 9. This rule describes the set of {4,5,10} objects. The rule’s factors’ set is {Ou:3, Wi:2}. The requirement is to perform the rules’ compression activity. The compression means to find and remove the rules that already contains the combination of factors {Ou:3, Wi:2}.

The “optimizeRulesBase” is called by line 07 in Figure 7. For every factor of the rule 9 it is required to get information about the set of rules that already describe the particular factor. It can be done using data from Figure 9. The factor Ou:3 is described by the set of rules {6,7,8} and the factor Wi:2 is described by the set of rules {2,7,8}. The combination of factors {Ou:3, Wi:2} is described by the intersection of the sets of rules of the individual factors: $\{2,7,8\} \cap \{6,7,8\} = \{7,8\}$. The result means, that the rule 7 and the rule 8 contains the combination of {Ou:3, Wi:2}. The rule 9 describes the objects more precisely. The rule 7 and rule 8 should be removed from the rules’ base. It is done by step 07 and step 09 in Figure 8.

The removed rules’ references should be also eliminated from “describedFactors” object. The new rule’s reference should be added to the factors of “describedFactors” object. This is done in steps 03, 06, 08, 10 in Figure 9.

In order to decrease the amount of work to do on the next stages it is also required to maintain the states of “describedObjects” and “rulesObjects”. This is done in steps 05, 13, 14 in Figure 10 and in steps 07, 08, 09 in Figure 11 respectively.

```
01: (3, {1})
02: (7, {1})
03: (12, {1})
04: (13, {1})
05: (5, {2,8})   (5, {2,8})   (5, {2,9})
06: (9, {2,4})
07: (1, {3,5})
08: (2, {3,5})
09: (11, {4})
10: (8, {5})
11: (6, {6})
12: (14, {6})
13: (4, {7})     (4, {7})     (4, {9})
14: (10, {7,8})  (10, {7,8})  (10, {9})
```

Figure 10. ZFFR algorithm's instance variable "describedObjects": Map[Int, Set]

```
01: (1, {3,7,12,13})
02: (2, {5,9})
03: (3, {1,2})
04: (4, {9,11})
05: (5, {1,2,8})
06: (6, {6,14})
07: (7, {4,10})   (7, {4,10})
08: (8, {5,10})   (8, {5,10})
09: (9, {4,5,10}
```

Figure 11. ZFFR algorithm's instance variable "ruleObjects" Map[Int, Set]

The result of “optimizeRulesBase” call is a compression of the rules’ base and maintained state of “describedFactors”, “describedObjects” and “ruleObjects” variables.

2.2.3 The principle: “don’t perform work that is already done”

Comparing with ZFFR algorithm’s implementation in [1] and [2], the current implementation requires the information of the set of objects mapped to the particular factor. Every factor, before adding to the processing stack, is checked for the objects’ described condition (step 19 Figure 7). If the factor’s objects are already described, then the factor will not be added to the stack and will not be processed. This reduces the amount of algorithm’s execution time. Author names such activity as a principle of “don’t perform work that is already done”. This principle is achieved by implementing the method “areObjectsDescribed”.

Suppose it is needed to check the requirement of processing the factor that contains the set of objects {2,1,8}. For every object it is necessary to get the sets of rules. Then the intersection of these sets of rules should be found. If the intersection is not empty, then these objects are already described, otherwise not.

From "describedObjects" (Figure 10) could be found that {3,5} ∩ {3,5} ∩ {5} = {5}. It means that these objects are already described by the rule number 5. And this factor shouldn't be added to the stack.

Suppose it is needed to check the requirement of processing the factor that contains the set of objects {12,13,5}. From the "describedObjects" (Figure 10) it could be found that {1} ∩ {1} ∩ {2,8} = {}. It means that these objects are not described yet by any rule. This factor should be added to the stack.

2.2.4 The principle: "don't assign work that is already assigned"

This principle of "don't assign work that is already assigned" is achieved by implementing the method "isFactorToVisit" (line 17 in Figure 7). This method is responsible for retrieving the information about factor's ids that are going to be processed. For example, if factor with id "Ou:2 Te:3" is already set to visit, then this factor will definitely not be added into the stack. If factor with id "Ou:1 Wi:1" is not yet set to visit, then it could be added to the stack to process, if "areObjectsDescribed" condition is successfully passed.

2.2.5 The dividing of algorithm's main flow into sub-flows technique

ZFFR algorithm's implementation in [1] and [2] could be used for calculation of all rules within one workflow. If a data set has a complicated structure, then one-flow implementation could take huge amount of time and sometimes even can lead to "out of memory" problem. The author's implementation is capable to perform the entire work using parallel flows. This functionality is achieved by implementing the method called "isProbableClassInside".

Suppose, it is required to define all rules for the class "P" for Quinlan data set [9]. The initial combinations of object-class map are presented in Figure 12. As a first example, imagine that it is given a factor with objects' set {3,4,6}. Should this factor be added to the stack to process? The Figure 12 shows the classes associated with these objects are

{“P”, “P”, “N”}. The rule for the required “P” class could be defined. That is why this factor should be added to the stack.

```
01: (1, “N”)
02: (2, “N”)
03: (3, “P”)
04: (4, “P”)
05: (5, “P”)
06: (6, “N”)
07: (7, “P”)
08: (8, “N”)
09: (9, “P”)
10: (10, “P”)
11: (11, “P”)
12: (12, “P”)
13: (13, “P”)
14: (14, “N”)
```

Figure 12. Quinlan data set: the initial combinations of object-class map

Should the factor with objects’ set of {8, 6, 2} be added to the stack, if it is required to find the rules for class “P”? The Figure 12 shows that the classes associated with these objects are {“N”, “N”, “N”}. The rule for the required “P” class couldn’t be defined if this factor will be added to stack. It means that this factor should be skipped and not processed by this workflow.

2.3 The method of object’s classification and accuracy calculation logic

The previous chapters’ methods, techniques and principles helps to generate rules that are used as a base for classification of unknown objects. In this part author will describe the classification’s method as well as accuracy calculation’s logic.

Suppose, the rules’ base for every data set class is generated and there is a data set that should be classified.

The rules’ base with the essential characteristics for Quinlan data set for the threshold 2 is presented in Table 4. Inside the table there are three essential characteristics of the rule that are involved in the algorithm of unknown objects’ set classification: “class_j”, “factors_j” and “excludes_j”. All three columns’ values are saved in JSON format. While classifying the test-objects author does not use all “excluded” factors. The set of

“excluded” factors of every rule contains only one arbitrary selected representative from each attribute (column).

rule_id	class_j	factors_j	excludes_j
1	[{"class": "N"}]	[{"outlook": "sunny"}, {"humidity": "high"}]	[{"temperature": "cool"}]
2	[{"class": "N"}]	[{"temperature": "hot"}, {"outlook": "sunny"}]	[]
3	[{"class": "N"}]	[{"outlook": "rain"}, {"windy": "TRUE"}]	[{"temperature": "hot"}]
4	[{"class": "P"}]	[{"humidity": "normal"}, {"windy": "FALSE"}]	[]
5	[{"class": "P"}]	[{"temperature": "cool"}, {"windy": "FALSE"}]	[{"outlook": "overcast"}]
6	[{"class": "P"}]	[{"temperature": "mild"}, {"humidity": "normal"}]	[{"outlook": "overcast"}]
7	[{"class": "P"}]	[{"outlook": "overcast"}]	[]
8	[{"class": "P"}]	[{"outlook": "sunny"}, {"humidity": "normal"}]	[{"temperature": "hot"}]
9	[{"class": "P"}]	[{"outlook": "rain"}, {"windy": "FALSE"}]	[{"temperature": "hot"}]

Table 4. Quinlan's data set rules' base, threshold 2

The test-objects are presented in Table 5. Test-object data is located inside the column named “dobject_j”. The format of the column’s value is JSON.

dobject_id	class_expected	dobject_j
1	[{"class": "N"}]	[{"outlook": "sunny"}, {"temperature": "hot"}, {"humidity": "high"}, {"windy": "FALSE"}]
2	[{"class": "P"}]	[{"outlook": "overcast"}, {"temperature": "hot"}, {"humidity": "high"}, {"windy": "FALSE"}]
3	[{"class": "N"}]	[{"outlook": "rain"}, {"temperature": "mild"}, {"humidity": "high"}, {"windy": "FALSE"}]

Table 5. Test-objects for Quinlan’s data set

The task is to classify the objects presented in Table 5, using the rules’ base presented in Table 4. The proposed algorithm of classification is the following. For every test-object

taken from Table 5 it is required to run the classification's sql-query written in Figure 13.

```
SELECT rules.class_j, count(rules.id)
FROM rules
WHERE rules.factors_j <@ {test-object.dobject_j}
      AND (rules.excludes_j = {empty_jsonb}
          OR (rules.excludes_j <> {empty_jsonb}
              AND NOT (rules.excludes_j <@ {test-object.dobject_j})))
GROUP BY 1;
```

Figure 13. Classification's sql-query

Sql-query presented in Figure 13 is written for PostgreSQL database using the principles from [11]. The element '<@' means 'the subset of'. The elements '{test-object.dobject_j}' and '{empty_jsonb}' are the placeholders for '[{"outlook": "sunny"}, {"temperature": "hot"}, {"humidity": "high"}, {"windy": "FALSE"}]' and '[]' values respectively.

The sql-query returns the number of rules that describes the particular test-object. It is essential to keep in mind that one test-object can be classified by rules that belong to different classes.

The count of described rules for every test-object are collected and saved separately in Table 6. This step is required to calculate the accuracy of classification.

accuracy	**class_expected**	**dobject_id**	**class_defined**	**N**	**P**
1	N	1	**N**	2	0
1	P	2	**P**	0	1
0	N	3	**P**	0	1
66,7%					

Table 6. The calculation of accuracy

Test-object 1 is successfully classified by rules 1 and 2, both belong to class "N". The total count of found rules is saved in first row under N (value 2) in Table 6. There are no rules of class "P" that describe the test-object 1. The value for the first row and column P is 0. "Class_defined" equals the name of the class's column, whose count of rules

value is maximum. In the case of test-object 1 “N” is chosen, because 2 for class “N” is greater than 0 for class “P”. The same logic is used for test-objects 2 and 3.

The calculation of accuracy is done, using the comparison of “class_expected” and “class_defined” values. If the values are equal, the “accuracy” column receives value of 1. Otherwise the “accuracy” column equals 0. The value of accuracy is calculated, using the equation 1 or 2.

$$Accuracy = \frac{Sum(\text{accuracy})}{Count(accuracy)} \quad (1)$$

$$Accuracy = \frac{Count(\{class_{expected}=class_{defined}\}\, testobject)}{Count(testobject)} \quad (2)$$

In the case of three test-objects, 2 objects were classified correctly and 1 object wasn’t. The accuracy percentage equals 66,7%.

2.4 Summary

In this chapter author has described the main data structures used in author’s implementation of ZFFR algorithm. The data structures “Factor” and “Rule” have been presented in an object-oriented way. The interpretation of “Rule” has been done also in JSON format what was making it possible to perform SQL queries to the rules’ base saved in PostgreSQL.

ZFFR algorithm’s structure together with rules’ technique of optimization have been described. The principles “don’t perform work that is already done” and “don’t assign work that is already assigned” together with the technique of how to divide ZFFR algorithm’s main flow into the sub-flows have been explained. The method of classification has been presented and explained based on the example.

3 Results

In Results chapter author will present the outputs of done research. The results will be mostly visualized using charts. Every output will be provided with the detailed comments.

This chapter consists of five subchapters. In the first part author will describe the training and testing data sets' structures as well as the statistical data received during the research process. In the next part the hypothesis check will be done. Using the graphical approach author will show the relationship between the number of rules generated and the rate of accuracy. After that will be shown the result of integration of multi-threaded approach into ZFFR algorithm and into the classification's method. The last part of the chapter will describe new problem that occurred during the implementation of the method of classification – the rules' conflicting problem.

3.1 The description of statistics received

The data set of 20000 instances from [3] was randomly permuted and divided into five subsets (K01, K02, K03, K04, K05) each by 4000 instances (Appendix 1 – "Letter recognition" data set). Then, using the logic presented in Table 7 the subsets were combined to get data sets for training and testing.

Id	Training data set	Testing data set
T01	K01, K02, K03, K04	K05
T02	K02, K03, K04, K05	K01
T03	K03, K04, K05, K01	K02
T04	K04, K05, K01, K02	K03
T05	K05, K01, K02, K03	K04

Table 7. The subsets combinations for training and test data sets

For every training data set for thresholds 1 to 5 were generated the bases of rules. Every testing data set was classified and estimated by rate of accuracy. The aggregated statistics is presented in appendices 2 – 31. The appendices 2 – 26 contain detailed

information for every combination of training and testing data sets. The appendices 27 – 31 contain average figures for detailed data presented in appendices 2 – 26.

Every table has the same calculated columns. TL means the number of letters observed. MA1 means the number of letters correctly classified under condition 1. AC1 means the accuracy rate calculated under condition 1. MA2 means the number of letters correctly classified under condition 2. AC2 means the accuracy rate calculated under condition 2. MA3 means the number of letters correctly classified under condition 3. AC3 means the accuracy rate calculated under condition 3. R means the number of rules found. RT means the time spent for calculating the rules in seconds. CT means the time spent for classifying the test objects in seconds.

A letter considered correctly classified under condition 1, if at least one correctly classified rule is found. Such accuracy rate is very important, because helps to identify the potential of ZFFR algorithm's classification function. Condition 1 is a simplified version of accuracy method described in subchapter 2.3. The definition of accuracy under condition 2 is equivalent to method described in subchapter 2.3. The detailed explanation of the condition 3 will be presented in the chapter 4.

The charts created in the next subchapters are done based on the statistics described above.

ZFFR algorithm and the implementation of method of classification were written using Scala programming language. Regarding ZFFR algorithm author had tested two prototypes where searching functionality for existence of "class", "zero factors" and "excluded factors" were realized for the first one in PostgreSql [12] and for the second one in KDB+ [13]. Both versions were rejected. Compared with pure implementation in Scala the time-efficient characteristics of the named prototypes were not appropriate.

The main advantage of the author's prototype to the one created by Liisa Jõgiste [2] includes the ability to find rules for each class in parallel, which could significantly decrease the overall time of execution. The other difference involves the "skill" to make a compression of rules at the time of rules' generating. There is no requirement to do the compression as a separate step like it was in [2]. The analyst receives already optimized rules' base without a need to perform an extra iteration.

The statistics is generated using MacBook Pro: 2,2 GHz Intel Core i7 processor; 16 GB 1600 MHz DDR3.

3.2 The accuracy result

The set hypothesis states that more rules are generated, more accurate classification's result could be obtained. To find out the existence of the relationship between the accuracy rate (AC2) and the number of rules generated (R) author will use graphical approach.

The relationship between accuracy rate and number of rules is presented in Figure 14.

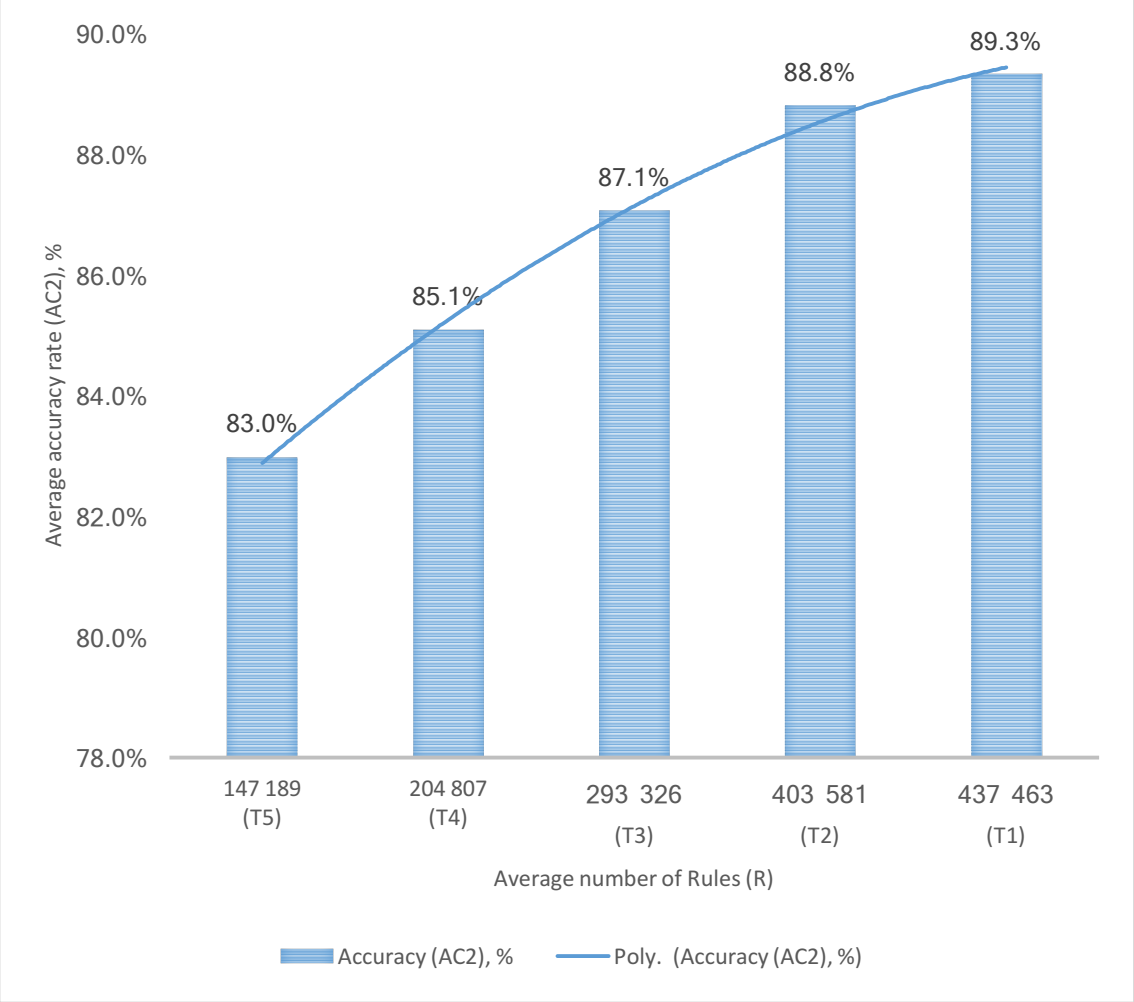

Figure 14. The relationship between accuracy rate and number of rules generated (A)

The data used for chart's creation is located in appendices 27 – 31. On the x-axis there are average numbers of rules generated for thresholds 5 to 1. On the y-axis there are rates of accuracy mapped for each threshold. The trend line emphasizes the strong

relationship between the number of rules and the rate of accuracy. The greater volume of rules leads to the greater percentage of accuracy rate.

Another evidence of the relationship's existence can be shown in Figure 15.

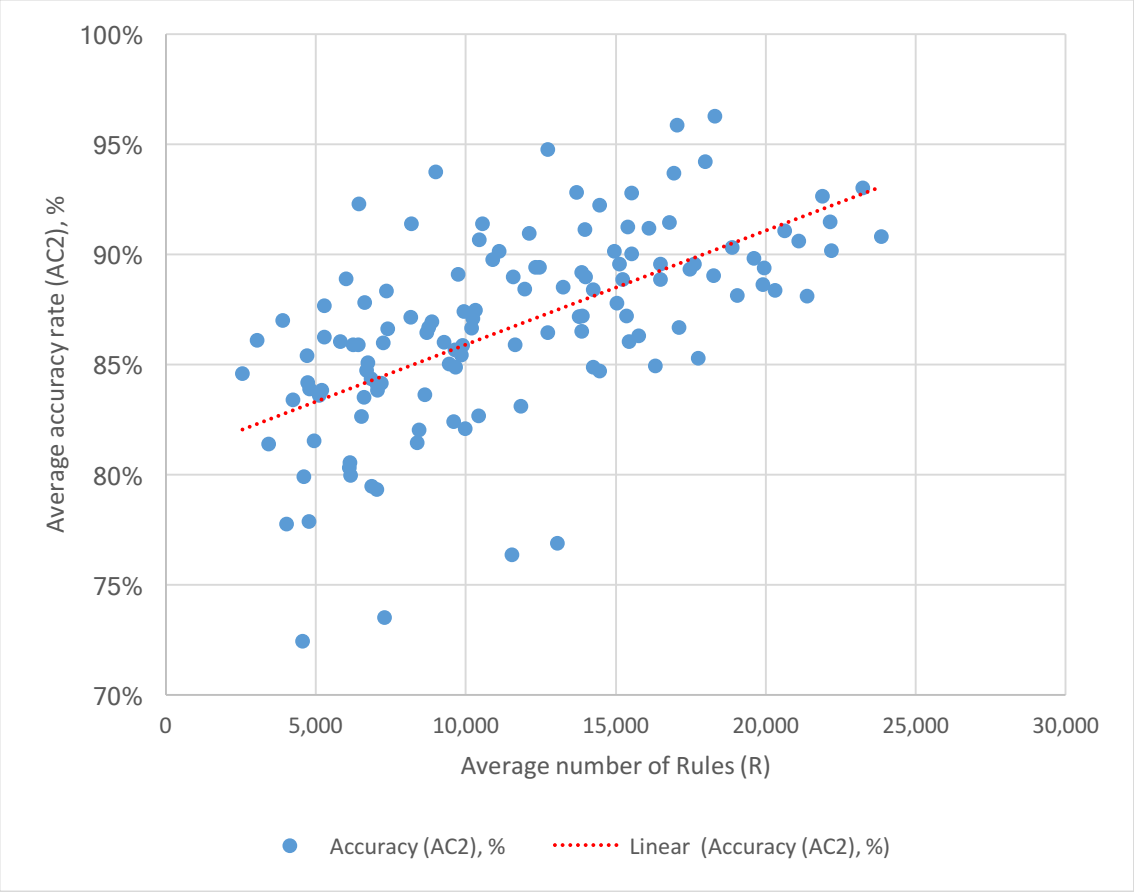

Figure 15. The relationship between accuracy rate and number of rules generated (B)

For chart's creation is used the same data located in appendices 27 – 31, but in this case not the aggregated totals, but detailed figures of every letter of every threshold from 5 to 1. The trend line also emphasizes the relationship between the number of rules and the rate of accuracy. The greater volume of rules leads to the greater percentage of accuracy rate.

The best result of accuracy is **89,3%**, achieved on 437 463 rules.

3.3 The result of influence of multi-threaded technique's integration in rules' generation algorithm

The results of the relationships between the number of rules and the rate of accuracy could not be achieved without multi-threaded technique used in implementation of ZFFR algorithm.

The total average amount of time for rules' generation under the threshold of 1 approximately equals 22069 seconds (6 hours). The rules could be generated after 6 hours only if ZFFR algorithm is run on appropriate hardware with significant amount of memory. The multi-threaded technique described in subchapter 2.2.5 decreased the amount of time spent to 1272 seconds (21 minutes) (Figure 16) and requirements to amount of memories used.

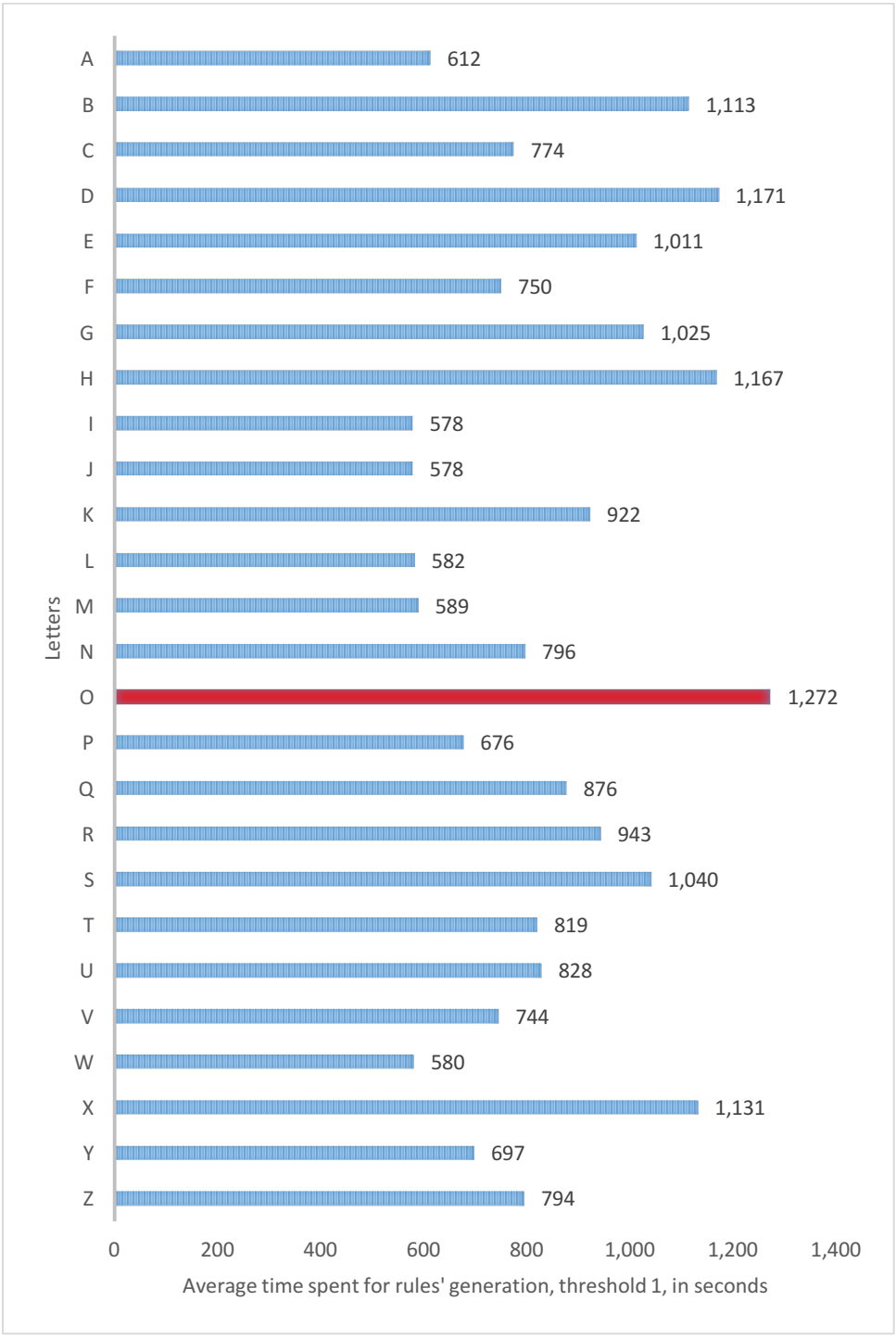

Figure 16. The average time spent for generation of rules

The received result for threshold 1 for multi-threaded approach is approximately 17,3 times faster than for single-threaded approach (22069/1272). The same rate for thresholds 5 to 1 is presented in Figure 17.

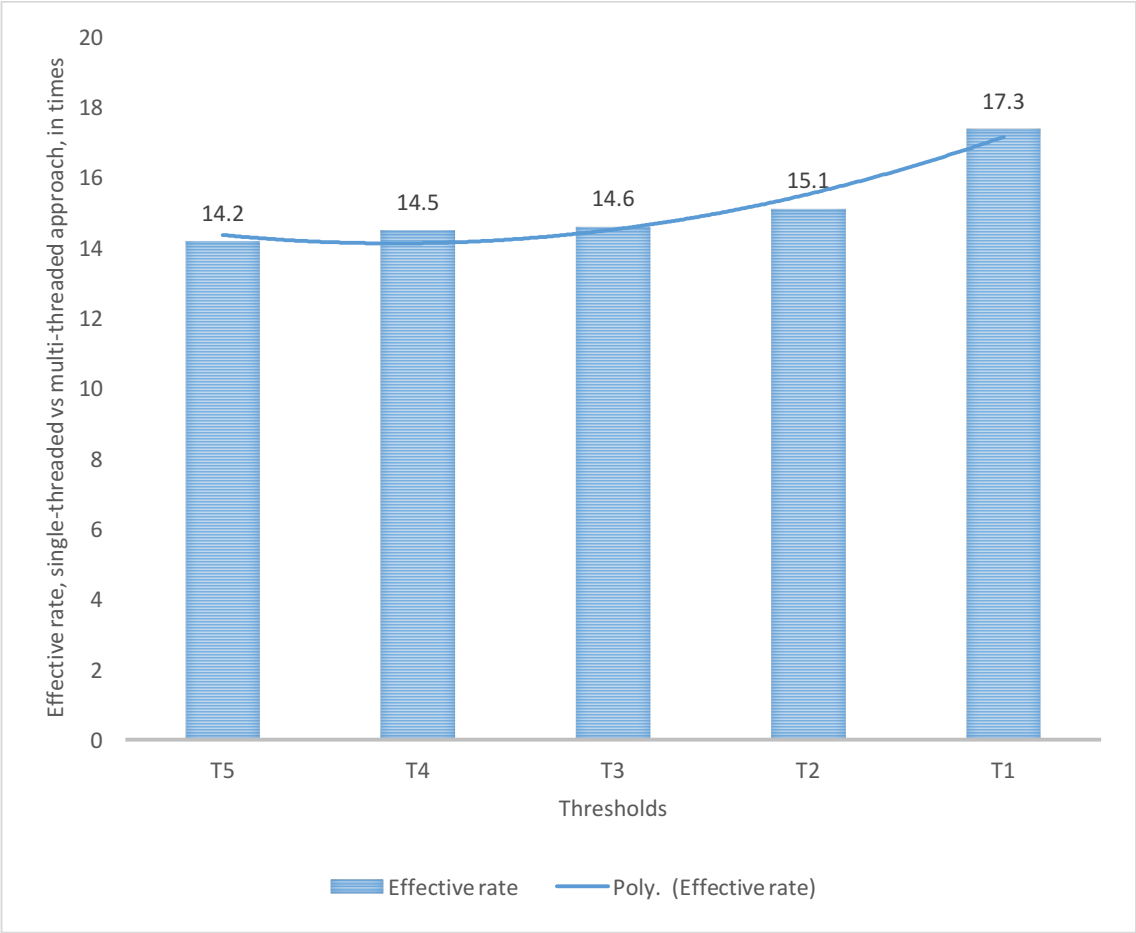

Figure 17. The effective rate, single-threaded vs multi-threaded approach (rules' generation)

The trend line also shows the clear relationship between effective rate and amount of work done. The lower the threshold, the greater the effective rate is achieved.

3.4 The result of influence of multi-threaded technique's integration in classification's algorithm

The method of classification described in 2.3 can be also run in parallel for each letter separately. The average time spent figures for classification on threshold 1 is presented in Figure 18.

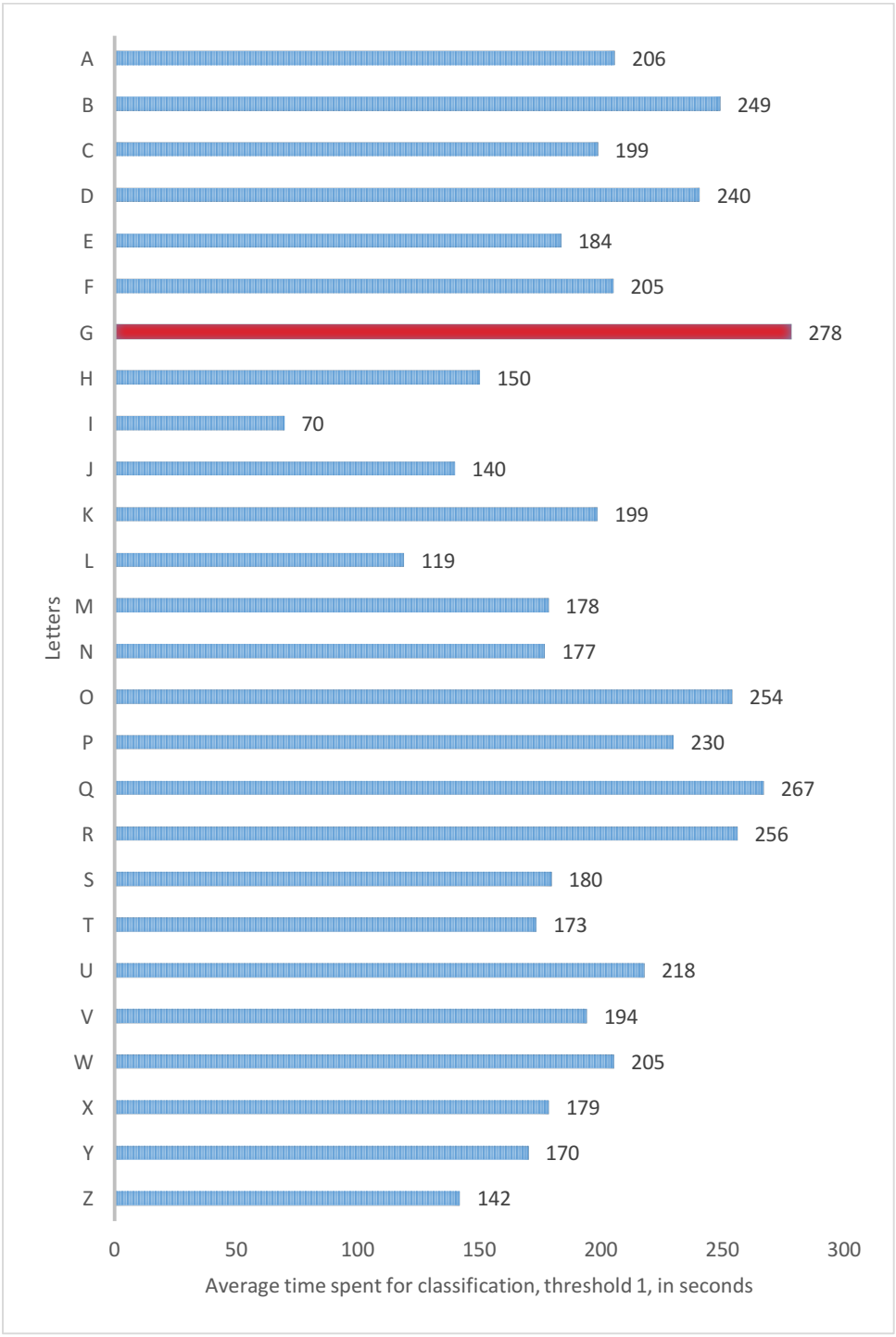

Figure 18. The average time spent for classification

The average amount of time for classification, using the rules received under the threshold 1 is approximately equals 5060 seconds (1,4 hours). The multi-threaded execution of technique described in subchapter 2.3 could decrease the amount of time spent to 278 seconds (5 minutes).

The received result for threshold 1 for multi-threaded approach is approximately 18,2 times faster than for single-threaded approach (5060/278). The same rate for thresholds 5 to 1 is presented in Figure 19.

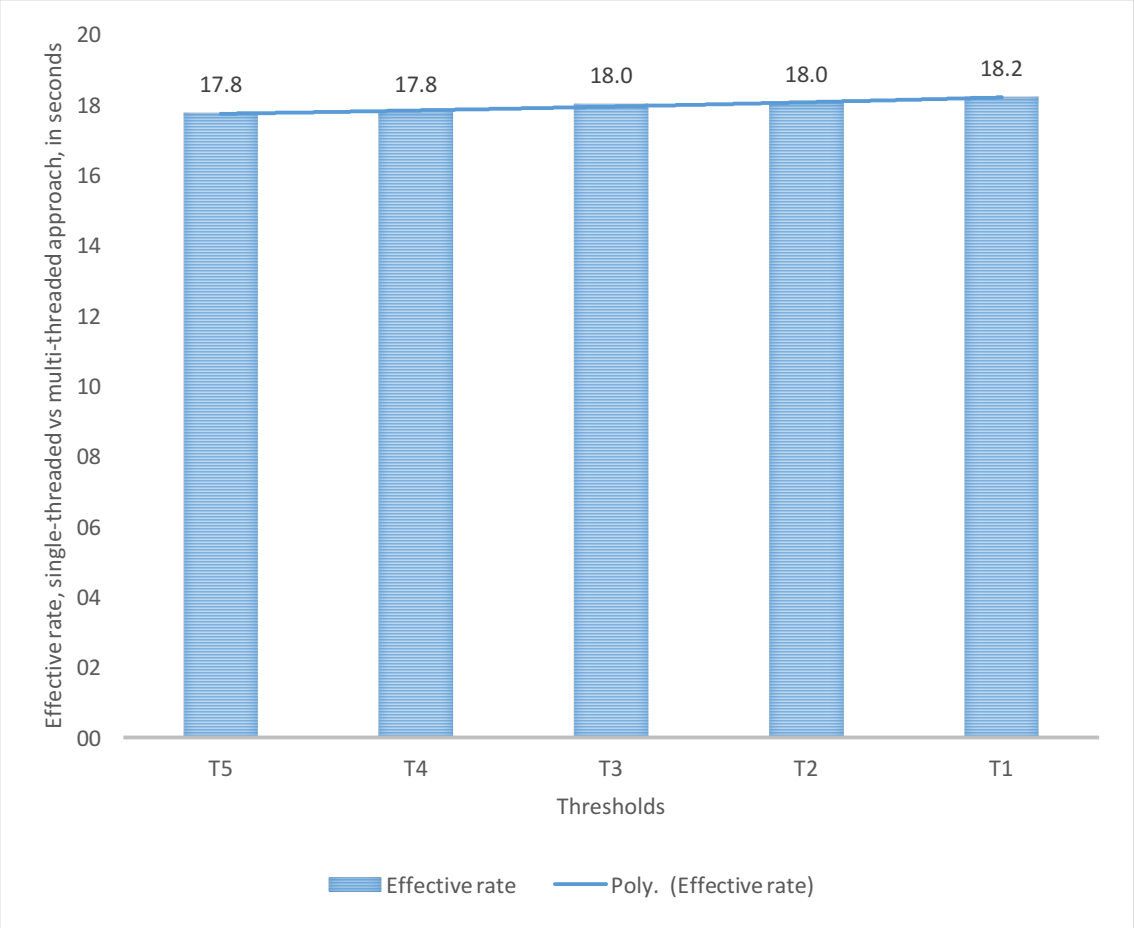

Figure 19. The effective rate, single-threaded vs multi-threaded approach (classification)

The trend line also shows the slight relationship between effective rate and amount of work done. The lower the threshold, the greater the effective rate is achieved.

3.5 Summary

In this chapter the main results have been described. The hypothesis: more rules are generated, more accurate classification's result could be obtained has been successfully tested and accepted. The prove has been presented in a graphical way. The maximum accuracy of 89,3% has been achieved on 437 463 rules generated on threshold 1.

The integration of multi-threaded techniques has been led to 17,3 (6 hours to 21 minutes) times win while rules' generating and 18,2 (1,4 hours to 5 minutes) times win while test-objects' classifying.

4 Discussion

In Discussion chapter author would like to comment the received results in more details. Three main aspects were chosen. The first one relates to the used classification's technique which could lead to the problem known as the rules conflict problem. The second one refers to multi-threaded technique used in ZFFR algorithm. The third part is aimed to check the reliability of the achieved results.

4.1 The rules conflict problem

The *potential* (AC1) of classification method described in subchapter 2.3 has an ability to find at least one rule that could correctly classify an unknown test-object. The potential of classification's method is presented in Figure 20 (**orange** bars).

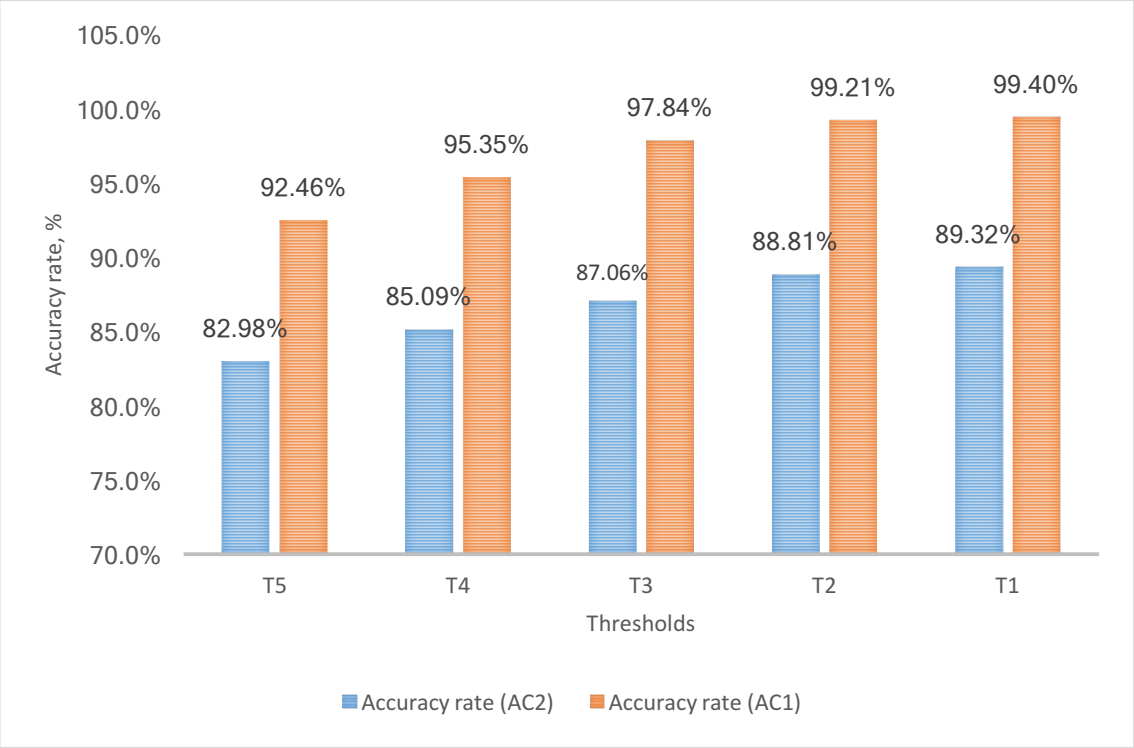

Figure 20. The potential rate of accuracy compared with rate of accuracy received by subchapter's 2.3 method

More rules are generated, the greater the potential rate of accuracy is received. On the threshold 1 it equals 99,4%. The classification's method is working and could define the unknown objects with very high rate of accuracy.

The figures of method of accuracy described in subchapter 2.3 (**blue** bars) are approximately 10% less than potential ones. The main reason here that the same test-object could be defined by rules belonged to different classes. If the count of "wrong" rules is greater than the count of "correct" rules, then the test-object will be classified incorrectly and the rate of accuracy will be less than potential.

This is not a new problem and is known as the rules conflict problem. One possible solution together with brief introduction to other methods could be seen in [14]. The main idea of most proposed solutions is to use some weights calculation to distinguish the "correct" class from the "incorrect" one.

Author proposes one other possible solution to the rules conflict problem. This solution could be used under the assumption that all unknown objects of data set are divided into the groups. In other words, the investigator knows that a group of some objects belongs to one class. The task here is to define this class.

Author describes the solution based on one test-object identification number 206 taken from T05 training/testing data set (subchapter 3.1). Rules' base is generated with threshold 5. Test-object 206 belongs to "S". Under the real circumstances there is no information about the class, but, as supposed, there is an information about the group of test-object, for example "GR01". To classify the test-object the next steps should be done.

On the first step it is essential to get the information about the rules that define the target test-object together with all other test-objects. The example of such query for one rule is presented in Table 8.

rule_id	rule_class	to_oid	to_group
6351007	S	9273	GR01
6351007	S	206	GR01
6351007	S	598	GR01
6351007	S	16866	GR01
6351007	S	912	GR02
6351007	S	6396	GR01
6351007	S	3617	GR02
6351007	S	10210	GR01

Table 8. Test-objects classification by rule's id 6351007

In Table 8 some test-objects belong to “GR01” and some to “GR02”. When this data is received it is needed to calculate two metrics. Metric 1 is a count of test-objects that belong to the same group as the target test-object (equals 6). Metric 2 is a count of all test-objects described by rule 6351007 (equals 8). The weight of the rule could be found as a rate between Metric 1 and Metric 2 (equals 6/8 or 0,75). On the next step the same calculation of weights should be done for all other rules that describe the target test-object 206. The aggregated data is presented in Table 9.

to_oid	to_group	rule_class	metric 1	metric 2	weight	rules_count
206	GR01	D	1	1	1,00	1
206	GR01	D	1	2	0,50	3
206	GR01	D	1	3	0,33	1
206	GR01	J	1	1	1,00	1
206	GR01	J	2	2	1,00	1
206	GR01	O	1	4	0,25	1
206	GR01	O	1	5	0,20	1
206	GR01	P	1	1	1,00	1
206	GR01	P	1	2	0,50	1
206	GR01	P	1	6	0,17	3
206	GR01	P	1	11	0,09	1
206	GR01	P	2	11	0,18	1
206	GR01	S	5	5	1,00	1
206	GR01	S	6	6	1,00	2
206	GR01	S	7	7	1,00	1
206	GR01	S	6	8	0,75	1
206	GR01	V	1	6	0,17	1

Table 9. Test-object 206 rules’ weights

On the final step the classification can be done based on received weights (Table 10).

to_oid	D	J	O	P	S	V
206	1,83	2,00	0,45	1,94	**3,75**	0,17

Table 10. Test-object 206 classification: weights based

The class definition is done based on the maximum value. For the target test-object the maximum value 3,75 belongs to class S. This is a correct classification.

The result of the technique described in subchapter 2.3 for the same test-object is presented in Table 11.

to_oid	D	J	O	P	S	V
206	5	2	2	**7**	5	1

Table 11. Test-object 206 classification: rules' count based

This data is received based on “rules_count” values of Table 9. The class definition is also done based on the maximum value. For the target test-object the maximum value 7 belongs to class P. This is an incorrect classification.

The logic of calculation of the rate of accuracy (AC3) for the classification method described in this subchapter is the same as was described in subchapter 2.3. The only difference here that for method AC2 the value is defined by the count of rules described the test-object. And for method AC3 the value is defined by the weight of rules described the test-object.

The overall result of classification done by method AC3 together with AC2 and AC1 methods is presented in Figure 21. The rate of accuracy based on method AC3 is still less than potential accuracy rate, but is approximately 3% higher than the same rate received by method AC2.

There is also a strong relationship between the number of rules and the rate of accuracy received by method AC3. The greater volume of rules leads to the greater percentage of accuracy rate. This can be seen in Figure 22.

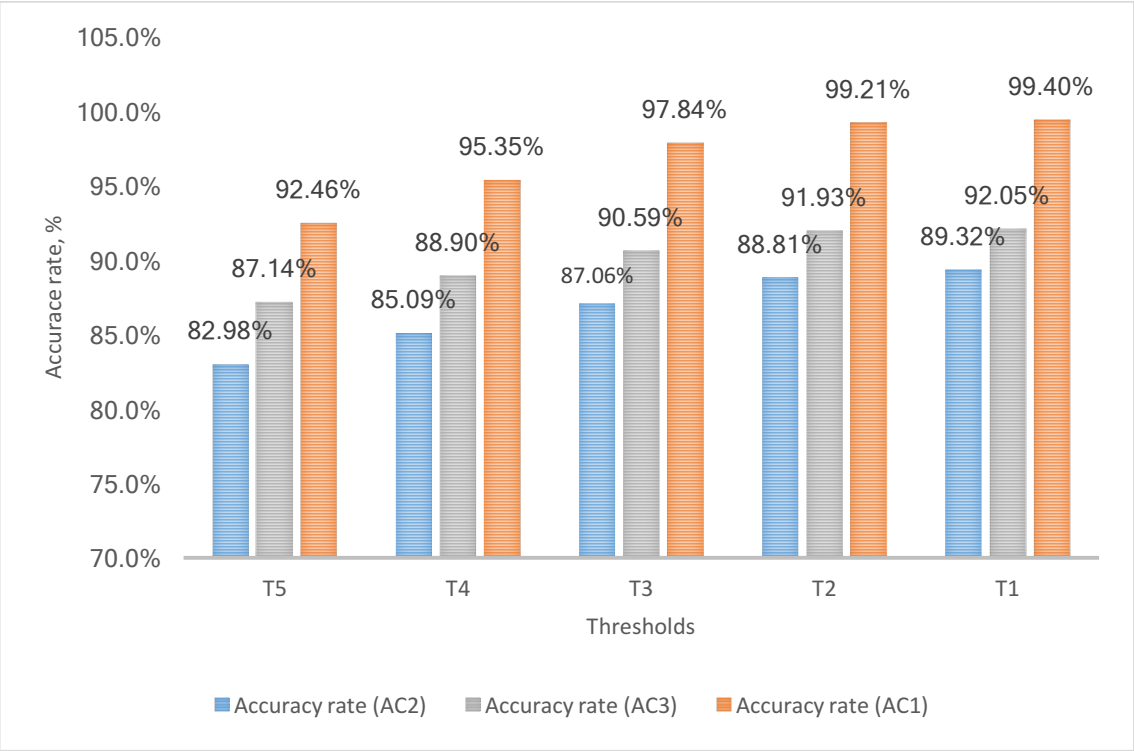

Figure 21. The comparison of classification's methods AC2, AC3, AC1

Figure 22. The relationship between accuracy rate and number of rules generated, method AC3

The proposed method of classification is based on assumption that unknown objects' data set is divided into the groups. The groups of test-objects can be also received by running algorithms. For example, dividing into the groups could be done using "minus technique" of Monotone System algorithms described in [15].

4.2 ZFFR algorithm's multi-threaded technique

The multi-threaded technique described in subchapter 2.2.5 could lead to a significant decrease of amount of time used for generating the rules. As was shown in the previous chapter the parallel technique could be approximately 17 times faster than the single-threaded solution and equals to the maximum amount of time spent by individual flow. In the case of letter "O" it was 1272 seconds (21 minutes).

The amount of time of 1272 seconds could be more decreased by doing the optimization of ZFFR algorithm's code. The time spent for generating rules for letter "O" by individual methods of ZFFR algorithm is presented in Figure 23.

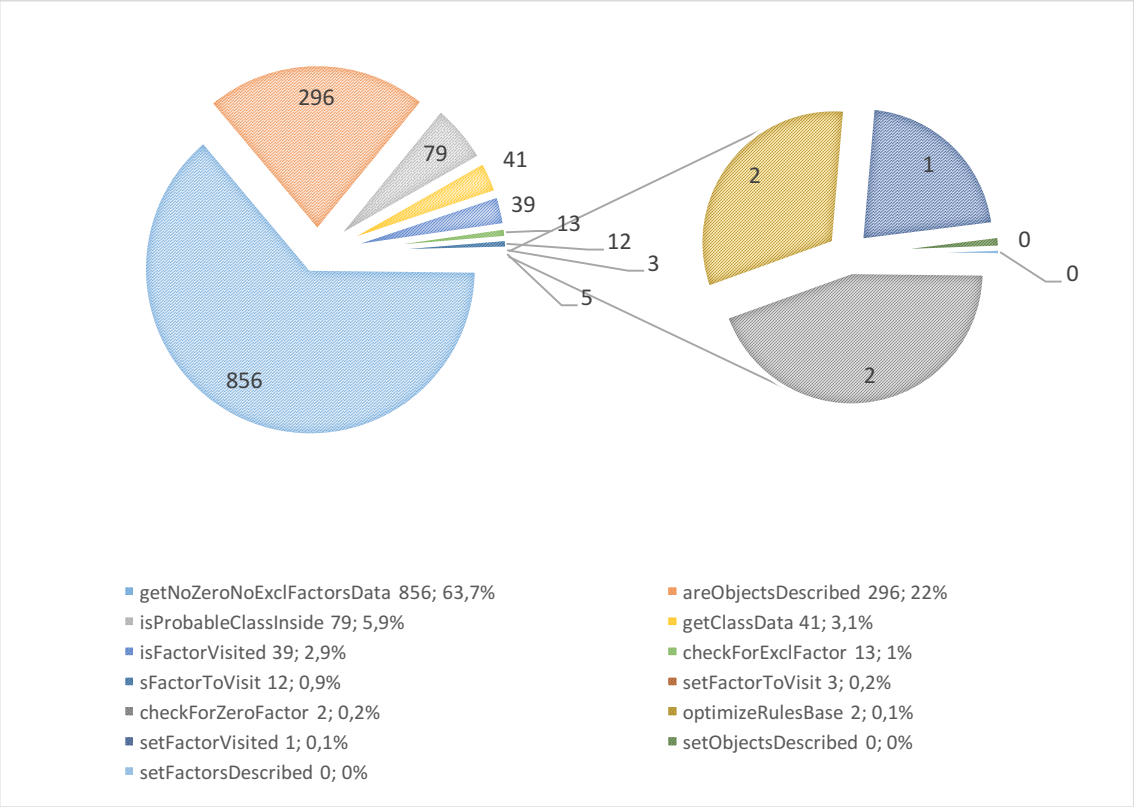

Figure 23. The time spent for generating rules for letter O by individual methods of ZFFR algorithm, in seconds

The most time-consuming methods are “getNoZeroNoExclFactorsData” and “areObjectsDescribed”. Approximately 67% and 23% of all time of algorithm is spent on execution of these methods respectively. The refactoring of these functions could lead to a significant decrease of overall ZFFR procedure.

4.3 The reliability of the achieved results

The aim of this subchapter is to check the reliability of the results achieved in chapter 3. There are three main questions that requires additional confirmation:

1. Does the set hypothesis: more rules lead to more accuracy - could be accepted for other data set as well?

2. Does the provided method of classification (2.3) lead to high rate of accuracy for other data sets as well?

3. Does the multi-threaded techniques for rules’ generating and for classification lead to time saving for other data sets as well?

Author will give the answers to these questions using another data from UCI Machine Learning Repository - “nursery” data set [16]. This data set was taken as a test-data set while the developing of ZFFR algorithm’s prototype in [2]. The original data set consists of 8 attributes and 12960 instances. The class distribution is presented in Table 12.

Class	**Number of instances**	**Number of instances, %**
not_recom	4320	33,3%
~~*recommend*~~	~~2~~	~~0,02%~~
~~*very_recom*~~	~~328~~	~~2,5%~~
priority	4266	32,9%
spec_prior	4044	31,2%

Table 12. "Nursery" data set, class distribution

Author decreased the number of instances participated in by removing the instances of two classes: recommend, very_recom. This was done in order to get almost equal distribution of class’s elements like it was in “letter recognition” data set [3]. The final data set includes 8 attributes, 12630 instances and 3 values of class.

4.3.1 The description of statistics received

The data set of 12630 instances from [16] was randomly permuted and divided into five subsets (K01a, K02a, K03a, K04a, K05a) by 2526 items (Appendix 32 – "Nursery" data set). Then, using the logic presented in Table 13 subsets were combined to get data sets for training and testing.

Id	Training data set	Testing data set
T01a	K01a, K02a, K03a, K04a	K05a
T02a	K02a, K03a, K04a, K05a	K01a
T03a	K03a, K04a, K05a, K01a	K02a
T04a	K04a, K05a, K01a, K02a	K03a
T05a	K05a, K01a, K02a, K03a	K04a

Table 13. The subsets combinations for training and test data sets, "nursery" data set

For every training data set for thresholds 1 to 5 were generated the bases of rules. Every testing data set was classified and estimated by rate of accuracy. The aggregated statistics is presented in appendices 33 – 37.

Every table has the same calculated columns. TL means the number of letters observed. MA1 means the number of letters correctly classified under condition 1. AC1 means the accuracy rate calculated under condition 1. MA2 means the number of letters correctly classified under condition 2. AC2 means the accuracy rate calculated under condition 2. MA3 means the number of letters correctly classified under condition 3. AC3 means the accuracy rate calculated under condition 3. R means the number of rules found. RT means the time spent for calculating the rules in seconds. CT means the time spent for classifying the test objects in seconds.

4.3.2 The accuracy result

In this subchapter author will get the answers on the first two questions about the acceptance of the hypothesis and about the value of accuracy. To find out the existence of the relationship between the accuracy rate (AC2) and the number of rules generated (R) author will use graphical approach.

The relationship between accuracy rate and number of rules is presented in Figure 24.

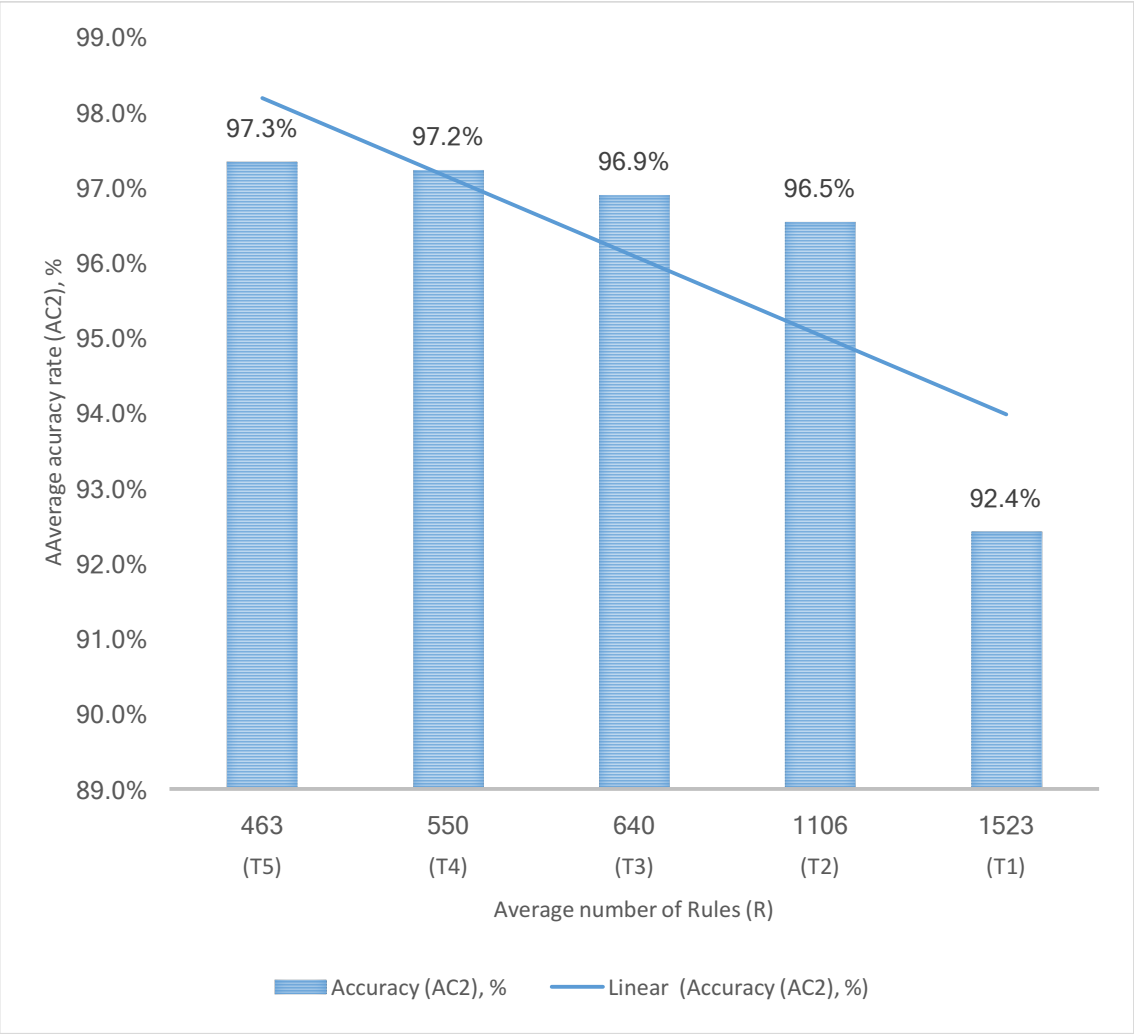

Figure 24. The relationship between accuracy rate and number of rules generated, "nursery" data set, AC2 (A)

The data used for chart's creation is located in appendices 33 – 37. On the x-axis there are average numbers of rules generated for thresholds 5 to 1. On the y-axis there are rates of accuracy mapped for each threshold. The trend line emphasizes the relationship between the number of rules and the rate of accuracy.

The set hypothesis for "nursery" data set for AC2 is not accepted. The result is totally opposite. Higher rate of accuracy is achieved by smaller number of rules.

Another evidence of the opposite relationship's existence can be shown in Figure 25.

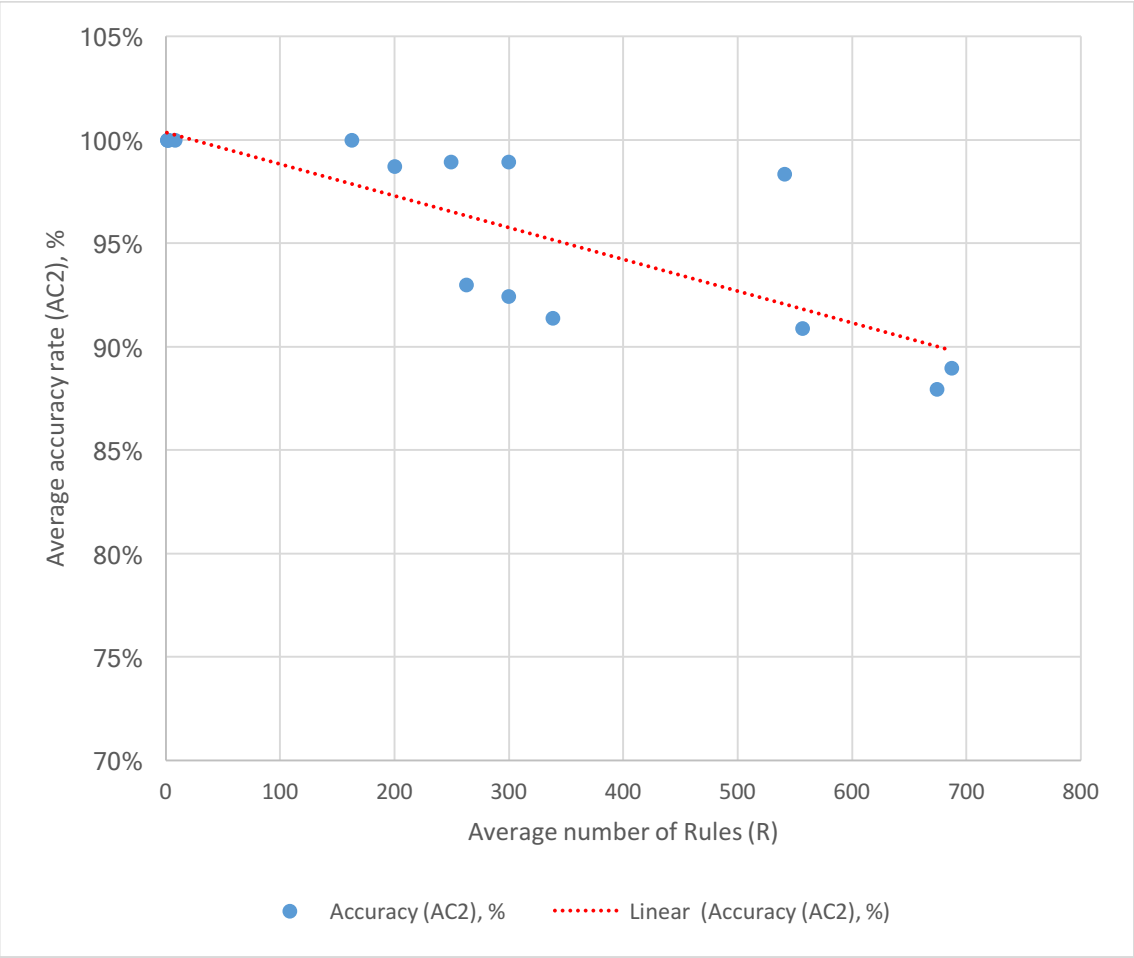

Figure 25. The relationship between accuracy rate and number of rules generated, "nursery" data set, AC2 (B)

For chart's creation is used the same data located in appendices 33 – 37, but in this case not the aggregated totals, but detailed figures of every value of a class for every threshold from 5 to 1. The trend line also emphasizes the opposite relationship between the number of rules and the rate of accuracy.

The best result of accuracy is **97,3%**, achieved on 463 rules.

The relationship between the accuracy rate (AC3) and the number of rules generated (R) is presented in Figure 26.

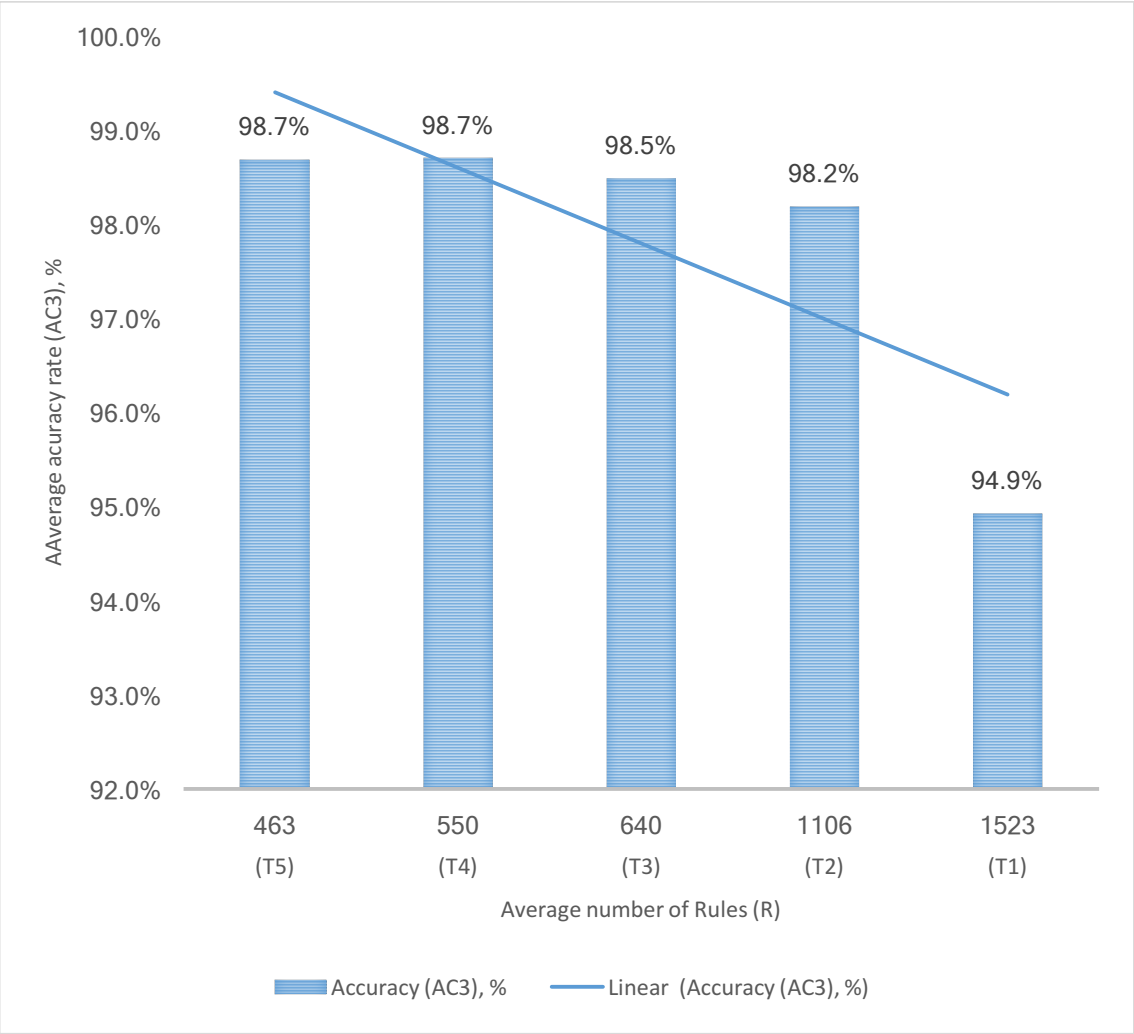

Figure 26. The relationship between accuracy rate and number of rules generated, "nursery" data set, AC3 (A)

The data used for chart's creation is located in appendices 33 – 37. On the x-axis there are average numbers of rules generated for thresholds 5 to 1. On the y-axis there are rates of accuracy mapped for each threshold. The trend line emphasizes the relationship between the number of rules and the rate of accuracy.

The set hypothesis for "nursery" data set for AC3 is also not accepted. The result is opposite. Higher rate of accuracy is achieved by smaller number of rules.

The detailed evidence of the relationship's existence is presented in Figure 27.

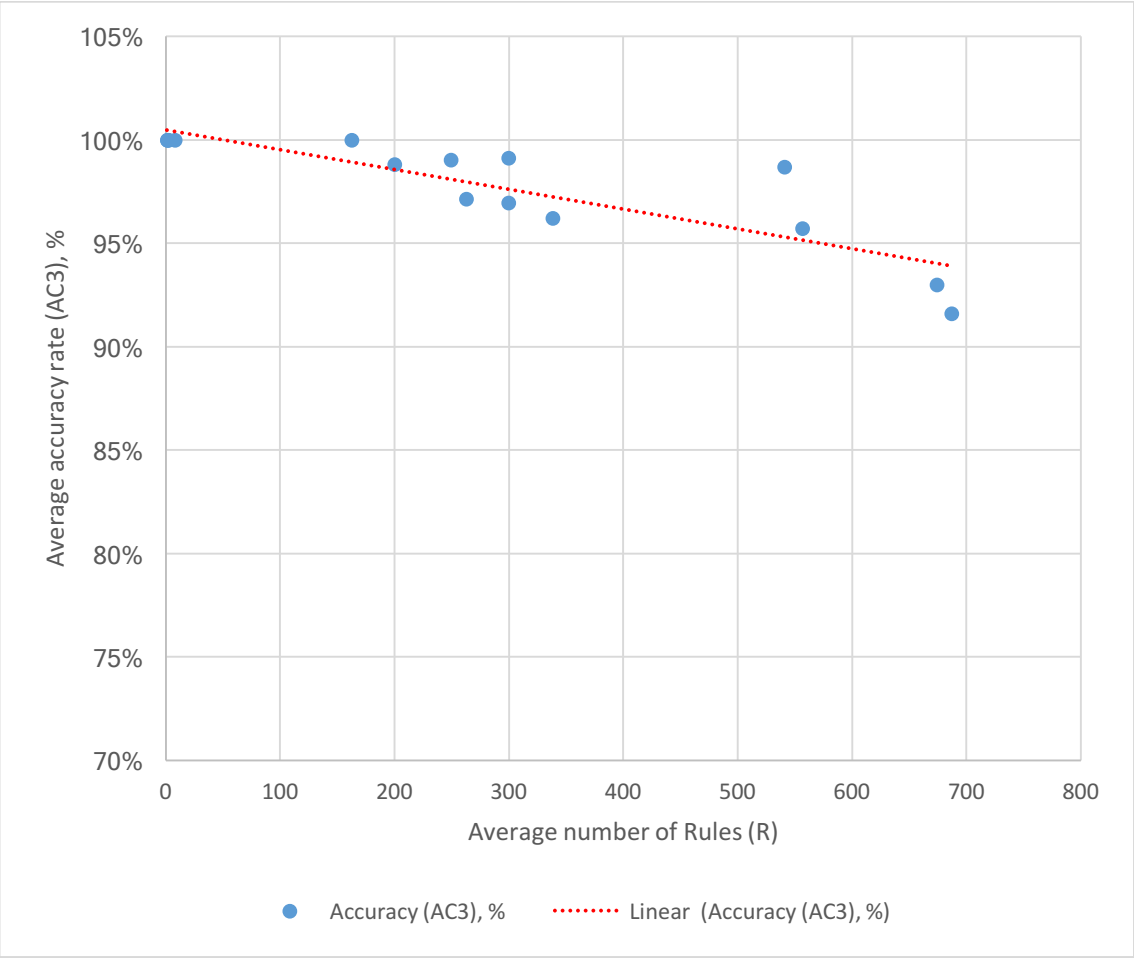

Figure 27. The relationship between accuracy rate and number of rules generated, "nursery" data set, AC3 (B)

For chart's creation is used the same data located in appendices 33 – 37, but in this case not the aggregated totals, but detailed figures of every value of a class of every threshold from 5 to 1. The trend line also emphasizes the opposite relationship between the number of rules and the rate of accuracy.

The best result of accuracy is **98,7%**, achieved on 463 rules.

The relationship between the accuracy rate (AC1) and the number of rules generated (R) is presented in Figure 28.

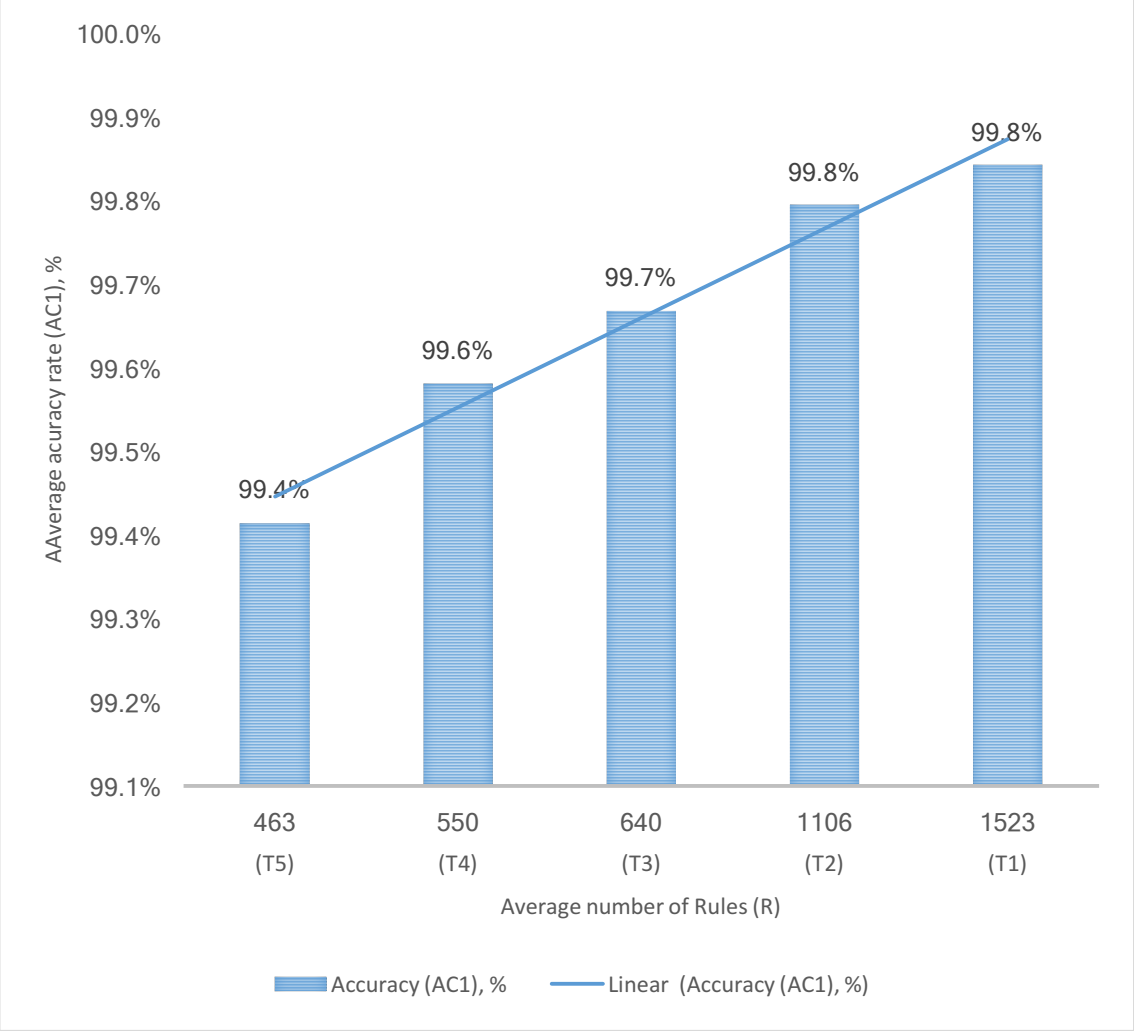

Figure 28. The relationship between accuracy rate and number of rules generated, "nursery" data set, AC1 (A)

The data used for chart's creation is located in appendices 33 – 37. On the x-axis there are average numbers of rules generated for thresholds 5 to 1. On the y-axis there are rates of accuracy mapped for each threshold. The trend line emphasizes the relationship between the number of rules and the rate of accuracy.

The set hypothesis for "nursery" data set for AC1 is accepted. Higher rate of accuracy is achieved by greater number of rules.

The best result of accuracy is **99,8%**, achieved on 1523 rules.

The set hypothesis: more rules lead to more accuracy for "nursery" data set is not accepted for techniques AC2 and AC3 but is accepted for AC1. Author accepts the set hypothesis based on AC1 result. More rules are generated more chances that a test-

object at least once will be classified correctly. The negative result of AC2 and AC3 is explained by rules' conflict problem.

The classification method for all cases provides the strong high results of accuracy of 97,3% and 98,7% and 99,8% for AC2 and AC3 and AC1 respectively.

4.3.3 The result of influence of multi-threaded technique's integration in rules' generation algorithm

The answer to the first part of the third question about the multi-threaded technique influence on rules' generating time will be found in this subchapter.

The total average amount of time for rules' generation under the threshold of 1 is approximately equals 39 seconds. The multi-threaded technique described in subchapter 2.2.5 decreased the amount of time spent to 14 seconds (Figure 29).

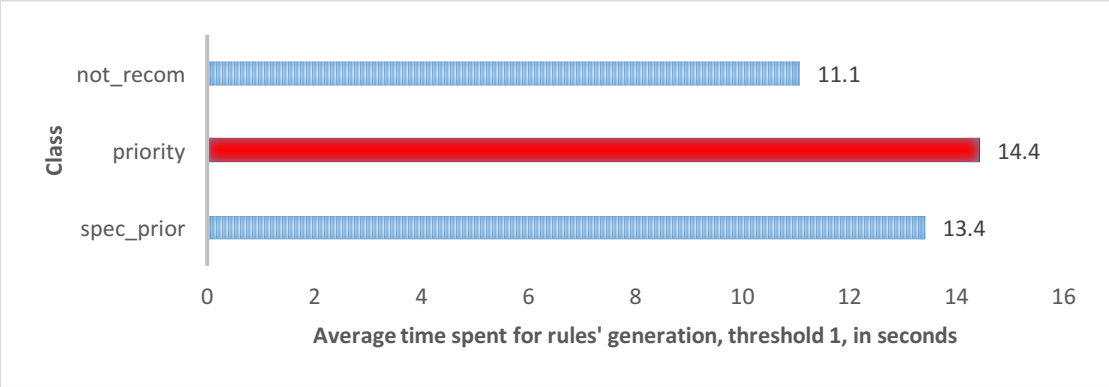

Figure 29. The average time spent for generation of rules, "nursery" data set

The received result for threshold 1 for multi-threaded approach is approximately 2,7 times faster than for single-threaded approach (39/14). The same rate for thresholds 5 to 1 is presented in Figure 30.

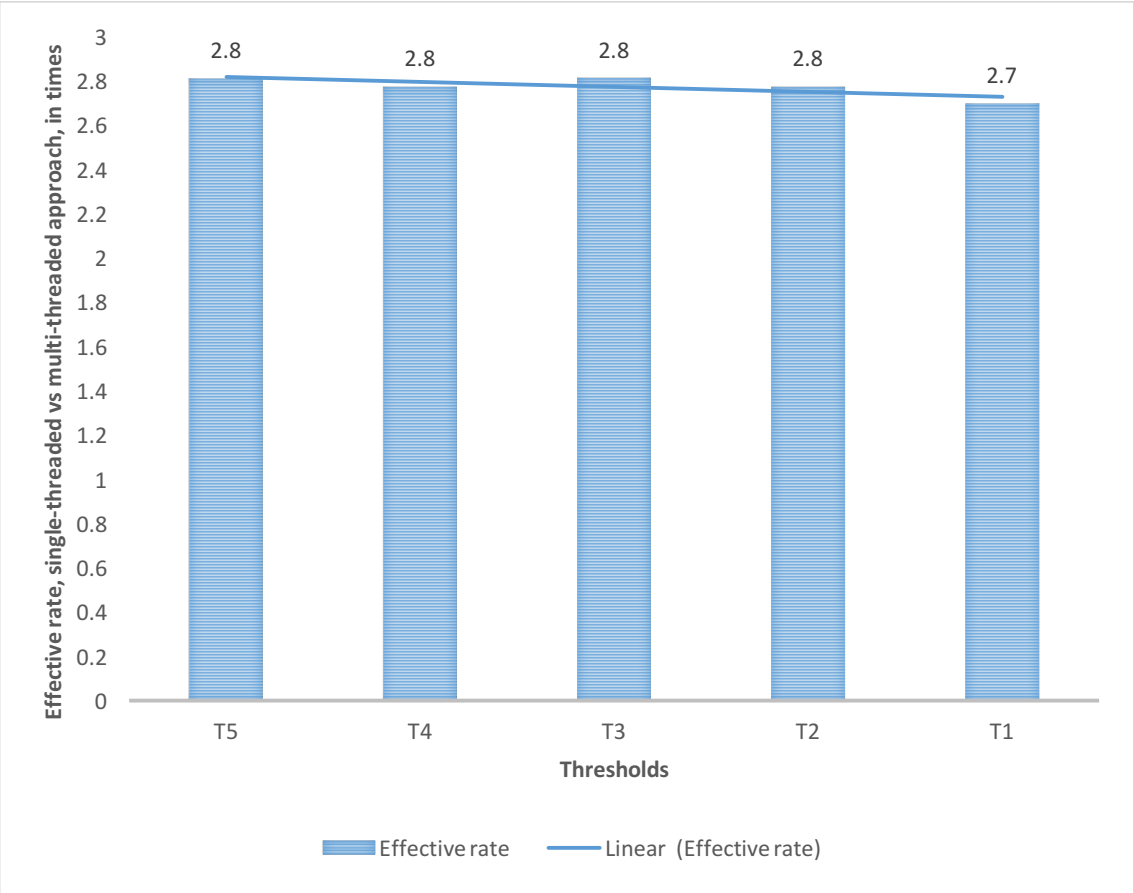

Figure 30. The effective rate, single-threaded vs multi-threaded approach, "nursery" data set (classification)

The achieved efficiency rate could be higher if the “nursery” data set includes more class’s values. The multi-threaded technique for rules’ generation leads to time saving result.

4.3.4 The result of influence of multi-threaded technique’s integration in classification’s algorithm

The answer to the second part of the third question about the multi-threaded technique influence on classification’s time will be found in this subchapter.

The total average amount of time classification under the threshold of 1 approximately equals 10 seconds. The multi-threaded execution of technique described in subchapter 2.3 could decrease the amount of time spent to 4 seconds (Figure 31).

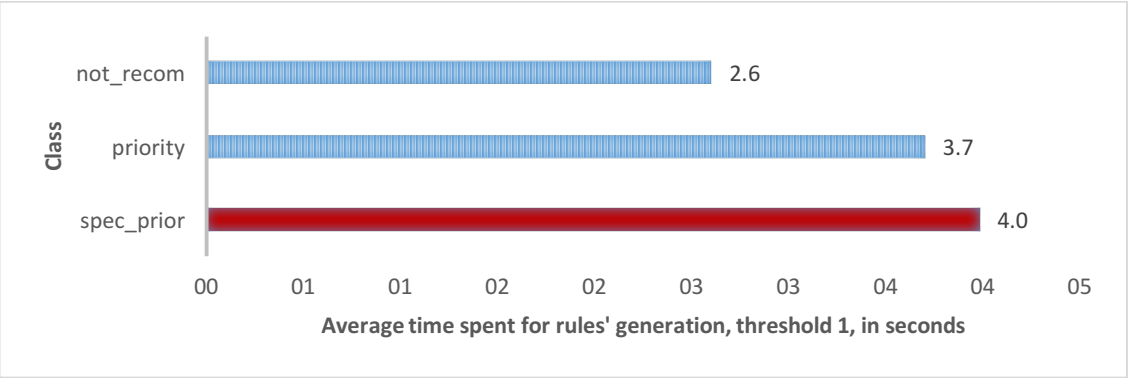

Figure 31. The average time spent for classification, "nursery" data set

The received result for threshold 1 for multi-threaded approach is approximately 2,6 times faster than for single-threaded approach (10/4). The same rate for thresholds 5 to 1 is presented in Figure 32.

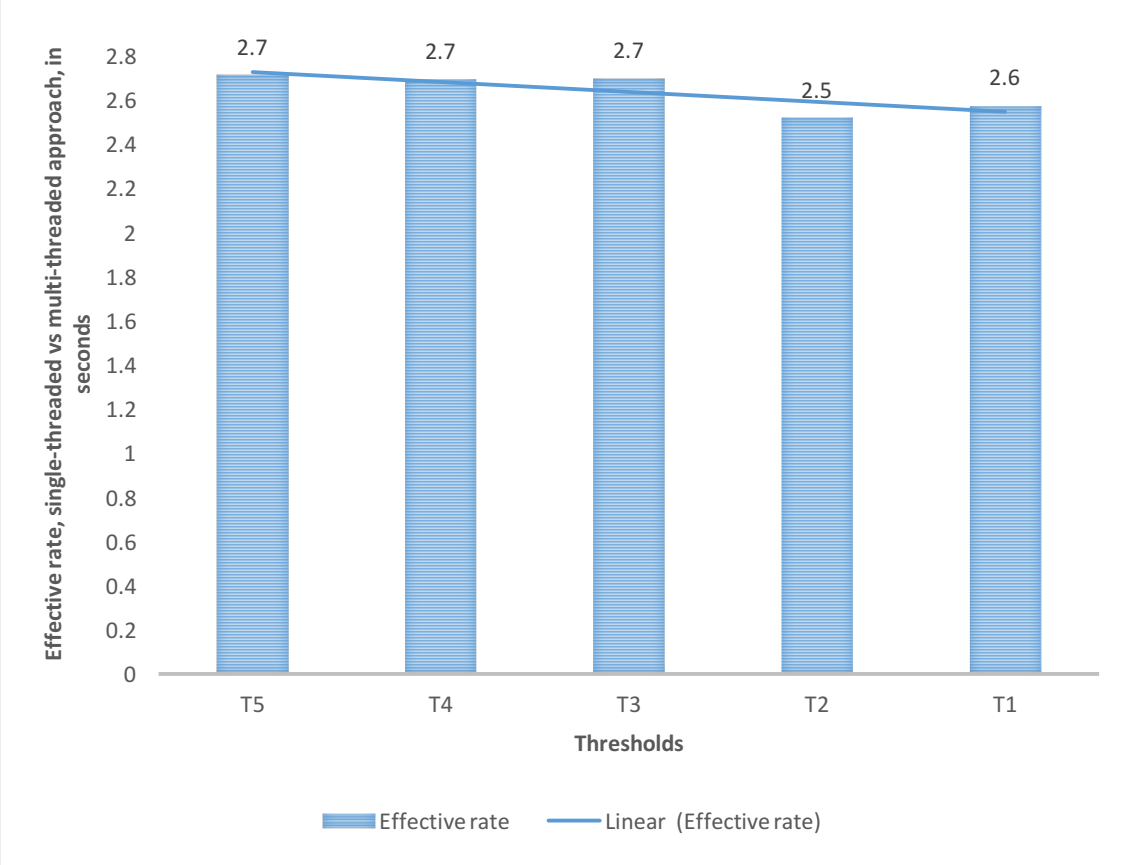

Figure 32. The effective rate, single-threaded vs multi-threaded approach, "nursery" data set (classification)

The achieved efficiency rate could be higher if the “nursery” data set includes more class’s values. The multi-threaded technique of classification leads to time saving result.

4.4 Summary

In this chapter author has explained the rules' conflict problem. Two new methods of classification have been proposed. The first method of classification (AC1) characterized by the ability to find at least one rule that could correctly classify an unknown test-object. This method could be used as a measure of rules' base potential to correctly classify the unknown data set. On the threshold 1 it equals 99,4%. The second method of classification (AC3) has been based on assumption that test-objects were divided into the groups. This method could be seen as author's way of solving the rules' conflict problem. The accuracy achieved on threshold 1 by using AC3 method was 92,05% which is greater than AC2 method's figure 89,32%.

As was shown in chapter 3, the time-efficient characteristics of rules' generation due to use of the multi-threaded techniques were significant. These characteristics can be more increased if to use more efficient coding techniques in methods of ZFFR algorithm. In this chapter author has pointed out the most time-consuming methods.

In order to check the reliability of the results achieved in chapter 3 author has made the same classifying work for other data set - "nursery" data set [16]. The set hypothesis: more rules lead to more accuracy for "nursery" data set has not been accepted for techniques AC2 and AC3 but has been accepted for AC1. Author has accepted the set hypothesis based on AC1 result. More rules are generated more chances that a test-object at least once will be classified correctly. Author has made a conclusion that the negative result of AC2 and AC3 has been explained by rules' conflict problem.

The classification method for all cases for "nursery" data set has provided the strong high results of accuracy of 97,3% and 98,7% and 99,8% for AC2 and AC3 and AC1 respectively.

The integration of multi-threaded techniques for "nursery" data set has been led to 2,7 (39 seconds to 14 seconds) times win while rules' generating and 2,6 (10 seconds to 4 seconds) times win while test-objects' classifying.

5 Summary

The main goal of the thesis was to develop a classification method acting on the rules received by ZFFR algorithm, estimate the developed method's accuracy and rank it compared to proposed benchmark table. Author set the hypothesis: more rules are generated, more accurately the worked out method is able to recognize the unknown objects. The testing of the hypothesis required making the improvements in ZFFR algorithm's time-effective characteristics in order to be able to generate maximum number of rules on the lowest levels of thresholds.

The new method of classification based on rules generated by ZFFR algorithm was created. The set hypothesis was successfully tested and accepted for "letter recognition" data set [3]: greater number of rules leads to more accurate results of classification.

The developed classification method was ranked compared to the results of accuracy received by other methods (Table 14).

Algorithm	**Accuracy (%)**
HSAC [4]	82,70
ZFFR AC2	**89,32**
Genetic programming [6]	92,00
ZFFR AC3	**92,05**
Neural networks [7]	94,13
FNNC [8]	95,67
LEBAI [5]	95,87

Table 14. The updated comparison of letters recognition accuracy

The accuracy of calculation method described in subchapter 2.3 (ZFFR AC2) on average equals 89,32%, which is higher than the result of HSAC algorithm. The method described in subchapter 4.1 (ZFFR AC3) on average provides 92,05%, which is higher than the result done by Genetic programming. These results are less than the result provided by LEBAI approach, but the estimated potential (99,40%) of classification

method (subchapter 4.1) requires to continue the investigation process. The new improved method of accuracy calculation, based on rules received by ZFFR algorithm, could lead to a higher result of accuracy, which means a new ranking position in the benchmark table.

The time-effective characteristics of ZFFR algorithm were improved. This could be done by integrating a "Factor" (subchapter 2.1.1) data structure inside the algorithm and refactoring the code using two main principles: "don't perform work that is already done" (subchapter 2.2.3) and "don't assign work that is already assigned" (subchapter 2.2.4). The significant amount of time spent by algorithm decreased by integrating the multi-threaded technique (subchapter 2.2.5). ZFFR algorithm spent approximately 17,3 times less of time for rules' generating on threshold 1 (subchapter 3.3).

The classification method proposed in subchapter 2.3 was also able to run in parallel workflows. The multi-threaded solution was approximately 18,2 times faster than a single-threaded implementation (subchapter 3.4).

The discussion done in subchapter 4.3 was aimed to confirm the achieved results based on other data set – "nursery" data set [16]. The set hypothesis was tested for AC2, AC3 and AC1 techniques. For AC2 and AC3 the opposite result was shown: higher rate of accuracy is achieved by smaller number of rules. AC1 result confirms the set hypothesis. Author accepts the set hypothesis based on AC1 result. More rules are generated more chances that a test-object at least once will be classified correctly. The negative result of AC2 and AC3 is explained by rules' conflict problem.

The classification method described in subchapter 2.3 (ZFFR AC2) provides the same strong result of accuracy for "nursery" data set of 97,3% and the method described in subchapter 4.1 (ZFFR AC3) provides also the strong result of accuracy of 98,7%.

Compared to "biological" algorithms used for solving the classification problem for "letter recognition" data set, ZFFR algorithm and its classification method described in this thesis has the clear advantage. The training part, or rules' generation, is purely done by ZFFR algorithm. An investigator should only run the algorithm on different thresholds to find out the best combination of rules for different values of class. In spite of the simplicity of training method, the rates of accuracy of classification's method are very high. Less amount of work done leads to very good results.

The received results could be possible due to work done in technical implementation of ZFFR algorithm, structural presentation of rule and parallelization.

The technical achievement 1. The use of "don't perform work that is already done" and "don't assign work that is already assigned" principles in ZFFR algorithm made possible to generate rules on threshold 1 for deeply structured data like "letter recognition" data set [3].

The technical achievement 2. The proposed JSON format rule's description made possible to perform SQL queries against rules' base saved in PostgreSQL.

The technical achievement 3. The proposed parallel techniques made possible to significantly improve the time-efficient characteristics of ZFFR algorithm and proposed classification techniques.

Author states that set targets of the thesis were achieved and the main problem was solved.

The possible future work 1. As was shown, the potential of classification function of ZFFR algorithm is very high. Inventing of a new effective solution to the rules conflict problem could increase the rate of accuracy and lead to a new ranking position in the benchmark table.

The possible future work 2. The classification method and the techniques of definition of accuracy were tested on two data sets "letter recognition" [3] and "nursery" [16]. To get more confidence on the received results it would be necessary to test the named methods on some other data set.

The possible future work 3. The finding out the best combination of rules generated on different thresholds very often requires acting on the lowest levels of thresholds, which put additional pressure to the time-efficient characteristics of ZFFR algorithm. The refactoring of ZFFR algorithm's internal methods could lead to significant decrease of amount of executed time and will make possible the generating of rules for data sets with highly complicated internal structure.

The possible future work 4. While testing the hypothesis for "nursery" data set [16], the result of the techniques AC2 and AC3 was negative: higher rate of accuracy is achieved

by smaller number of rules. Author explained the negative result by rules' conflict problem. The nature of the negative result provided by AC2 and AC3 techniques requires additional investigation.

Kokkuvõte

Magistritöö põhiline eesmärk oli välja töötada ZFFR algoritmiga leitud reeglitel põhinev klassifitseerimismeetod, hinnata meetodi täpsust ning võrrelda saadud täpsuseid teiste klassifitseerimismeetodite tulemustega. Autor püstitas hüpoteesi: mida rohkem reegleid, seda täpsem klassifitseerimistulemus. Hüpoteesi testimine nõudis ZFFR algoritmi reeglite genereerimiskiiruse parandamist, kuna reeglite genereerimisparameeter – sageduspiir – pidi olema võimalikult madal.

Töö tulemusena loodi uus klassifitseerimismeetod. Püstitatud hüpotees sai edukalt testitud ja vastu võetud (andmekogumil "letter recognition data set" [3]): mida rohkem reegleid saadud, seda täpsem klassifitseerimistulemus on saavutatud.

Loodud klassifitseerimismeetodi täpsust võrreldi teiste klassifitseerimismeetodite tulemustega (Table 15).

Algoritm	**Täpsus (%)**
HSAC *[4]*	82,70
ZFFR AC2	**89,32**
Genetic programming *[6]*	92,00
ZFFR AC3	**92,05**
Neural networks *[7]*	94,13
FNNC *[8]*	95,67
LEBAI *[5]*	95,87

Table 15. Uuendatud pingerea tabel, täpsus ("letter recognition data set")

Peatükis 2.3 kirjeldatud klassifitseerimismeetodi täpsuse arvutamisloogika (ZFFR AC2) andis tulemuseks keskmiselt 89,32%, mis on kõrgem kui HSAC meetodil saavutatud tulemus. Peatükis 4.1 kirjeldatud klassifitseerimismeetodi täpsuse arvutamisloogika (ZFFR AC3) saavutas tulemuse 92,05%, mis on kõrgem kui meetodiga "Genetic programming" saavutatud tulemus. Saadud tulemused on madalamad kui meetodiga LEBAI saavutatud tulemused. Autor on veendunud, et täpsuse tulemust saab veelgi

parandada. See järeldus on tehtud klassifitseerimismeetodi arvutatud potentsiaali põhjal (99,40%, peatükk 4.1).

Oluliselt parandati ZFFR algoritmi reeglite genereerimise kiirust. Selleks integreeriti andmestruktuur "Factor" (peatükk 2.1.1) ning rakendati kaht printsiipi: "ära tee tööd, mis on juba tehtud" (peatükk 2.2.3) ja "ära määra tööd, mis on juba määratud" (peatükk 2.2.4). Oluline ajavõit saavutati tänu paralleliseerimistehnikale (peatükk 2.2.5). ZFFR algoritm kulutas ligikaudselt 17,3 korda vähem aega reeglite genereerimiseks sageduspiiriga 1 (peatükk 3.3).

Paralleliseerimistehnikat rakendati ka klassifitseerimismeetodi suhtes, mida kirjeldati peatükis 2.3. Tänu sellele tehnikale kulutas klassifitseerimismeetod ligikaudselt 18,2 korda vähem aega (peatükk 3.4).

Saadud tulemusi analüüsiti peatükis 4.3. Selleks viidi läbi eksperiment teisel andmekogumil "nursery data set" [16]. Püstitatud hüpoteesi testiti tehnikate AC2, AC3, AC1 korral. AC2 ja AC3 korral olid tulemused vastupidised: täpsem klassifitseerimistulemus saavutati väiksema reeglite arvuga. AC1 tulemus kinnitas püstitatud hüpoteesi. Autor võtab püstitatud hüpoteesi vastu AC1 tehnika tulemuse põhjal. Mida rohkem reegleid, seda suurem tõenäosus, et test-objekt klassifitseeritakse vähemalt ühel juhul õigesti. AC2 ja AC3 meetodite negatiivset tulemust seletab autor "reeglite konflikti" probleemiga.

Peatükis 2.3 kirjeldatud klassifitseerimismeetod (ZFFR AC2) näitab kõrget täpsust (97,3%) ka "nursery" andmekogumi jaoks. Sama kõrge täpsuse tulemus (98,7%) on saavutatud ka ZFFR AC3 tehnika kasutamisel (peatükk 4.1).

ZFFR algoritm ja seotud klassifitseerimstehnikad omavad selget eelist võrreldes "bioloogiliste" algoritmidega. Treenimise osa, nimelt reeglite genereerimine, tehakse ZFFR algoritmi poolt. Algoritm peab olema käivitatav, kasutades erinevaid sageduspiiri väärtuseid, et saada kõige parem reeglite hulk klassi iga väärtuse jaoks. Vaatamata treenimise meetodi lihtsusele, on magistritöös pakutud klassifitseerimismeetodi täpsus väga kõrge.

Magistritöös saadud eksperimentaalsed tulemused tuginevad suurel määral järgmistele tehnilistele teostustele.

Tehniline teostus 1. Printsiipide “Ära tee tööd, mis on juba tehtud” ja “ära määra tööd, mis on juba määratud” kasutamine võimaldas reeglite arvutamise sageduspiiril 1 keeruka sisestruktuuriga andmekogumile nagu “letter recognition” [3].

Tehniline teostus 2. Reegli kirjeldamiseks kasutatud JSON-formaat võimaldas luua ja rakendada SQL-keeles loodud päringuid PostgreSQL andmebaasis salvestatud reeglitele.

Tehniline teostus 3. Pakutud paralleliseerimistehnikad võimaldasid märgatavalt vähendada ZFFR algoritmi ajakulu reeglite genereerimisel ja samuti ka ajakulu klassifitseerimismeetoditel.

Autor väidab, et magistritöös püstitatud eesmärgid on saavutatud ja tehnilised probleemid on lahendatud.

Võimalik tulevane töö 1. Töös on näidatud, et pakutud klassifitseerimismeetodi potentsiaal on väga kõrge. Uue parema lahenduse leidmine “reeglite konflikti” probleemile suudaks tõsta täpsusmäära veelgi ning viia klassifitseerimismeetodi uuele positsioonile pingerea tabelis.

Võimalik tulevane töö 2. Klassifitseerimismeetod ja täpsuse määratlused on testitud andmekogumitel “letter recognition” [3] ning “nursery” [16]. Meetodi töökindlus vajab kinnitust veel teiste andmekogumite töötlemisel.

Võimalik tulevane töö 3. Parema reeglite kombinatsiooni otsimine nõuab tegutsemist kõige madalamatel sageduspiiridel, mis omakorda esitab kõrged nõudmised ZFFR algoritmi reeglite genereerimise kiirusele. ZFFR algoritmi sisemeetodite parandamine võib viia märgatavale ajavõidule ning seeläbi võimaldab reeglite genereerimist keeruka struktuuriga andmekogumil.

Võimalik tulevane töö 4. Läbi viidud eksperiment andmekogumil “nursery data set” [16] näitas vastupidiseid tulemusi AC2 ja AC3 tehnikate korral: täpsem klassifitseerimistulemus saavutati väiksema reeglite arvuga. Autor seletab AC2 ja AC3 meetodite negatiivset tulemust “reeglite konflikti” probleemiga. Saadud negatiivne tulemus nõuab lisauurimist ja täpsemat põhjendust.

References

[1] G. Lind and R. Kuusik, “Algorithm for Finding Zero Factor Free Rules,” *Advances in Intelligent Systems and Computing,* vol. 391, pp. 421-435, 2015.

[2] L. Jõgiste, Prototyping of Zero-factor based DA, Master's Thesis, Tallinn: University of Technology, 2014.

[3] “UCI Machine Learning Repository,” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Letter+Recogmition.

[4] P. W. Frey and D. J. Slate, “Letter Recognition Using Holland-Style Adaptive Classifiers,” *Machine Learning,* vol. 6, no. 2, pp. 161-182, 1991.

[5] C. Liang, L. Peng, Y. Hong and J. Wang, “An English Letter Recognition Algorithm Based Artificial Immune,” *Advanced in Neural Networks,* vol. 5553, pp. 371-379, 2009.

[6] M. Ahluwalia and L. Bull, “Coevolving Functions in Genetic Programming,” *System Architecture,* vol. 47, 2001.

[7] G. Daqi, X. Chao and N. Guiping, “Combinative Neural-network-based Classifiers for Optical Handwritten Character and Letter Recognition,” in *International Joint Conference on Neural Networks, 3*, 2003.

[8] T. C. Fogarty, “First nearest neigbor classification on Frey and Slate's letter recognition problem,” *Machine Learing,* vol. 9, no. 4, pp. 387-388, 1992.

[9] J. R. Quinlan, “Induction of Decision Trees,” *Machine Learning 1,* pp. 81-106, 1986.

[10] “Introducing JSON,” [Online]. Available: www.json.org.

[11] “PostgreSQL: JSON Functions and Operations,” [Online]. Available: https://www.postgresql.org/docs/current/static/functions-json.html.

[12] “PostgreSql,” [Online]. Available: https://www.postgresql.org/.

[13] “KDB+,” [Online]. Available: https://kx.com/.

[14] T. Treier, “A New Effective Approach for Solving the Rules Conflict Problem,” in *2011 Third Pacific-Asia Conference on Circuits, Communications and System (PACCS)*, 2011.

[15] L. Võhandu, R. Kuusik, A. Torim, E. Aab and G. Lind, “Some algorithms for data table (re)ordering using Monotone Systems,” in *The 5th WSEAS International Conferation on Artificial Intelligence*, Madrid, 2006.

[16] “UCI Machine Learning Repository,” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Nursery.

Appendix 1 – “Letter recognition” data set

letter	K01	K02	K03	K04	K05	Total
A	161	161	152	163	152	**789**
B	154	155	158	150	149	**766**
C	148	149	151	148	140	**736**
D	164	152	178	143	168	**805**
E	156	149	168	146	149	**768**
F	158	150	161	174	132	**775**
G	154	151	150	158	160	**773**
H	145	155	154	132	148	**734**
I	144	163	157	149	142	**755**
J	162	140	161	143	141	**747**
K	143	144	131	161	160	**739**
L	145	155	138	146	177	**761**
M	161	143	162	150	176	**792**
N	163	142	160	147	171	**783**
O	144	170	140	149	150	**753**
P	151	166	161	157	168	**803**
Q	139	152	176	160	156	**783**
R	171	166	129	152	140	**758**
S	136	152	152	160	148	**748**
T	146	166	144	173	167	**796**
U	180	150	174	158	151	**813**
V	158	141	146	165	154	**764**
W	142	159	149	157	145	**752**
X	161	153	155	163	155	**787**
Y	167	151	159	154	155	**786**
Z	147	165	134	142	146	**734**
	4000	**4 000**	**4 000**	**4 000**	**4 000**	**20 000**

Appendix 2 – Statistics for T01, threshold 1

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	152	100,0%	145	95,4%	147	96,7%	18 325	591	209
B	149	149	100,0%	131	87,9%	134	89,9%	21 460	1 101	266
C	140	139	99,3%	126	90,0%	127	90,7%	17 854	749	206
D	168	167	99,4%	146	86,9%	146	86,9%	19 672	1 133	257
E	149	149	100,0%	136	91,3%	140	94,0%	15 501	1 006	201
F	132	132	100,0%	116	87,9%	124	93,9%	18 362	753	220
G	160	159	99,4%	147	91,9%	147	91,9%	24 127	993	**284**
H	148	148	100,0%	118	79,7%	120	81,1%	13 196	1 089	140
I	142	142	100,0%	127	89,4%	125	88,0%	5 958	596	65
J	141	139	98,6%	120	85,1%	121	85,8%	12 929	596	133
K	160	160	100,0%	139	86,9%	147	91,9%	16 615	939	181
L	177	175	98,9%	147	83,1%	159	89,8%	9 315	589	106
M	176	176	100,0%	164	93,2%	166	94,3%	14 793	617	156
N	171	170	99,4%	151	88,3%	157	91,8%	14 529	868	156
O	150	148	98,7%	132	88,0%	137	91,3%	21 399	**1 362**	245
P	168	166	98,8%	144	85,7%	147	87,5%	19 833	717	212
Q	156	156	100,0%	146	93,6%	153	98,1%	22 699	892	237
R	140	139	99,3%	129	92,1%	134	95,7%	22 937	936	244
S	148	147	99,3%	130	87,8%	133	89,9%	15 181	1 079	157
T	167	166	99,4%	154	92,2%	157	94,0%	15 124	857	155
U	151	151	100,0%	139	92,1%	142	94,0%	19 719	887	211
V	154	152	98,7%	143	92,9%	147	95,5%	16 376	777	176
W	145	144	99,3%	139	95,9%	143	98,6%	18 560	633	198
X	155	155	100,0%	140	90,3%	148	95,5%	15 589	1 149	173
Y	155	155	100,0%	141	91,0%	138	89,0%	14 883	713	158
Z	146	145	99,3%	130	89,0%	136	93,2%	11 849	809	132
	4 000	**3 981**	**99,5%**	**3 580**	**89,5%**	**3 675**	**91,9%**	**436 785**	**1 362**	**284**

Appendix 3 – Statistics for T02, threshold 1

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	161	100,0%	153	95,0%	151	93,8%	18 001	598	211
B	154	153	99,4%	136	88,3%	141	91,6%	21 190	1 087	251
C	148	148	100,0%	133	89,9%	136	91,9%	17 810	792	206
D	164	163	99,4%	139	84,8%	149	90,9%	19 970	1 153	234
E	156	155	99,4%	137	87,8%	136	87,2%	15 604	987	183
F	158	157	99,4%	137	86,7%	148	93,7%	17 908	766	205
G	154	153	99,4%	140	90,9%	144	93,5%	23 643	1 002	276
H	145	142	97,9%	109	75,2%	115	79,3%	13 256	1 168	156
I	144	138	95,8%	127	88,2%	128	88,9%	6 097	561	72
J	162	160	98,8%	146	90,1%	153	94,4%	12 396	561	143
K	143	142	99,3%	123	86,0%	123	86,0%	17 518	908	203
L	145	145	100,0%	132	91,0%	138	95,2%	10 406	578	120
M	161	161	100,0%	152	94,4%	153	95,0%	15 349	589	178
N	163	163	100,0%	138	84,7%	144	88,3%	15 041	793	175
O	144	142	98,6%	126	87,5%	130	90,3%	21 402	**1 331**	255
P	151	149	98,7%	136	90,1%	137	90,7%	20 649	703	236
Q	139	139	100,0%	133	95,7%	134	96,4%	24 244	913	**280**
R	171	171	100,0%	161	94,2%	164	95,9%	21 784	937	252
S	136	134	98,5%	122	89,7%	126	92,6%	16 457	1 035	192
T	146	145	99,3%	126	86,3%	131	89,7%	15 498	819	176
U	180	180	100,0%	166	92,2%	164	91,1%	18 014	792	212
V	158	154	97,5%	141	89,2%	139	88,0%	17 172	692	197
W	142	142	100,0%	138	97,2%	138	97,2%	18 485	581	210
X	161	160	99,4%	154	95,7%	156	96,9%	15 438	1 133	181
Y	167	167	100,0%	153	91,6%	159	95,2%	15 268	711	174
Z	147	147	100,0%	136	92,5%	138	93,9%	12 258	818	142
	4 000	**3 971**	**99,3%**	**3 594**	**89,9%**	**3 675**	**91,9%**	**440 858**	**1 331**	**280**

Appendix 4 – Statistics for T03, threshold 1

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	160	99,4%	149	92,5%	155	96,3%	18 105	619	207
B	155	154	99,4%	133	85,8%	141	91,0%	21 367	1 121	226
C	149	149	100,0%	135	90,6%	140	94,0%	17 517	744	180
D	152	152	100,0%	140	92,1%	146	96,1%	20 111	1 177	242
E	149	149	100,0%	132	88,6%	135	90,6%	15 404	996	181
F	150	150	100,0%	135	90,0%	141	94,0%	18 007	768	208
G	151	151	100,0%	136	90,1%	141	93,4%	24 164	1 050	**282**
H	155	152	98,1%	116	74,8%	124	80,0%	12 601	1 200	149
I	163	161	98,8%	143	87,7%	146	89,6%	5 947	575	70
J	140	140	100,0%	129	92,1%	134	95,7%	12 545	602	145
K	144	143	99,3%	124	86,1%	133	92,4%	16 758	872	194
L	155	154	99,4%	140	90,3%	147	94,8%	10 245	579	118
M	143	143	100,0%	134	93,7%	139	97,2%	16 145	600	183
N	142	142	100,0%	120	84,5%	134	94,4%	15 528	806	179
O	170	169	99,4%	153	90,0%	160	94,1%	19 533	**1 221**	232
P	166	166	100,0%	139	83,7%	143	86,1%	19 842	616	227
Q	152	152	100,0%	137	90,1%	143	94,1%	23 262	861	269
R	166	166	100,0%	150	90,4%	158	95,2%	20 799	881	242
S	152	149	98,0%	125	82,2%	133	87,5%	16 417	984	191
T	166	163	98,2%	144	86,7%	154	92,8%	15 097	747	174
U	150	150	100,0%	135	90,0%	139	92,7%	19 040	779	221
V	141	139	98,6%	127	90,1%	129	91,5%	17 318	735	200
W	159	159	100,0%	154	96,9%	157	98,7%	18 170	567	207
X	153	153	100,0%	138	90,2%	139	90,8%	15 256	1 133	178
Y	151	151	100,0%	138	91,4%	145	96,0%	15 313	692	175
Z	165	165	100,0%	154	93,3%	151	91,5%	11 671	772	135
	4 000	**3 982**	**99,6%**	**3 560**	**89,0%**	**3 707**	**92,7%**	**436 162**	**1 221**	**282**

Appendix 5 – Statistics for T04, threshold 1

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	152	100,0%	145	95,4%	145	95,4%	18 290	625	210
B	158	158	100,0%	143	90,5%	147	93,0%	21 190	1 107	251
C	151	151	100,0%	133	88,1%	138	91,4%	17 140	781	198
D	178	177	99,4%	163	91,6%	167	93,8%	19 052	1 221	224
E	168	167	99,4%	146	86,9%	157	93,5%	14 464	1 035	170
F	161	158	98,1%	130	80,7%	139	86,3%	17 482	716	201
G	150	150	100,0%	138	92,0%	142	94,7%	24 681	1 038	**288**
H	154	152	98,7%	118	76,6%	129	83,8%	12 583	1 150	149
I	157	156	99,4%	141	89,8%	147	93,6%	5 794	581	68
J	161	157	97,5%	144	89,4%	152	94,4%	12 054	557	139
K	131	130	99,2%	117	89,3%	117	89,3%	17 923	956	207
L	138	135	97,8%	113	81,9%	116	84,1%	10 499	577	121
M	162	161	99,4%	150	92,6%	153	94,4%	15 712	576	179
N	160	160	100,0%	139	86,9%	146	91,3%	14 960	722	174
O	140	140	100,0%	132	94,3%	135	96,4%	21 601	**1 264**	257
P	161	161	100,0%	147	91,3%	148	91,9%	20 209	709	231
Q	176	176	100,0%	163	92,6%	166	94,3%	22 890	889	265
R	129	129	100,0%	122	94,6%	122	94,6%	23 083	1 018	267
S	152	151	99,3%	130	85,5%	135	88,8%	15 584	1 079	178
T	144	143	99,3%	128	88,9%	134	93,1%	15 652	836	179
U	174	172	98,9%	151	86,8%	152	87,4%	19 036	831	223
V	146	144	98,6%	133	91,1%	136	93,2%	16 775	745	193
W	149	149	100,0%	142	95,3%	142	95,3%	18 294	535	208
X	155	154	99,4%	144	92,9%	140	90,3%	15 269	1 104	182
Y	159	159	100,0%	135	84,9%	146	91,8%	15 035	674	172
Z	134	132	98,5%	124	92,5%	127	94,8%	12 565	785	145
	4 000	**3 974**	**99,4%**	**3 571**	**89,3%**	**3 678**	**92,0%**	**437 817**	**1 264**	**288**

Appendix 6 – Statistics for T05, threshold 1

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	163	163	100,0%	151	92,6%	157	96,3%	17 156	626	191
B	150	149	99,3%	132	88,0%	136	90,7%	21 688	1 150	250
C	148	147	99,3%	132	89,2%	135	91,2%	17 706	803	202
D	143	143	100,0%	131	91,6%	133	93,0%	20 923	1 172	245
E	146	144	98,6%	119	81,5%	126	86,3%	15 791	1 031	182
F	174	173	99,4%	141	81,0%	151	86,8%	16 936	745	189
G	158	158	100,0%	141	89,2%	146	92,4%	22 604	1 044	260
H	132	132	100,0%	103	78,0%	107	81,1%	13 567	**1 228**	156
I	149	143	96,0%	133	89,3%	134	89,9%	6 221	577	74
J	143	143	100,0%	129	90,2%	132	92,3%	12 286	574	140
K	161	159	98,8%	137	85,1%	147	91,3%	16 719	936	207
L	146	145	99,3%	127	87,0%	129	88,4%	10 467	586	130
M	150	148	98,7%	135	90,0%	140	93,3%	15 620	562	194
N	147	146	99,3%	139	94,6%	138	93,9%	15 057	792	201
O	149	149	100,0%	139	93,3%	142	95,3%	21 475	1 183	281
P	157	155	98,7%	143	91,1%	144	91,7%	20 954	636	243
Q	160	160	100,0%	149	93,1%	154	96,3%	23 000	826	**283**
R	152	151	99,3%	131	86,2%	137	90,1%	22 070	942	275
S	160	159	99,4%	138	86,3%	143	89,4%	15 100	1 024	180
T	173	173	100,0%	156	90,2%	163	94,2%	14 781	837	182
U	158	156	98,7%	143	90,5%	144	91,1%	18 593	850	222
V	165	162	98,2%	155	93,9%	159	96,4%	16 211	772	205
W	157	157	100,0%	151	96,2%	156	99,4%	18 019	585	203
X	163	162	99,4%	142	87,1%	154	94,5%	15 458	1 135	179
Y	154	153	99,4%	137	89,0%	141	91,6%	15 130	695	172
Z	142	142	100,0%	124	87,3%	126	88,7%	12 162	787	156
	4 000	**3 972**	**99,3%**	**3 558**	**89,0%**	**3 674**	**91,9%**	**435 694**	**1 228**	**283**

Appendix 7 – Statistics for T01, threshold 2

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	152	100,0%	143	94,1%	147	96,7%	17 206	249	222
B	149	149	100,0%	132	88,6%	134	89,9%	19 894	402	274
C	140	139	99,3%	126	90,0%	127	90,7%	16 719	267	211
D	168	167	99,4%	144	85,7%	148	88,1%	17 978	435	230
E	149	149	100,0%	135	90,6%	140	94,0%	14 030	363	177
F	132	132	100,0%	117	88,6%	123	93,2%	16 890	252	232
G	160	158	98,8%	146	91,3%	146	91,3%	22 440	370	**308**
H	148	147	99,3%	117	79,1%	119	80,4%	11 648	438	151
I	142	141	99,3%	122	85,9%	127	89,4%	5 236	256	61
J	141	138	97,9%	119	84,4%	121	85,8%	12 082	188	139
K	160	160	100,0%	136	85,0%	146	91,3%	14 943	282	174
L	177	175	98,9%	145	81,9%	157	88,7%	8 473	180	95
M	176	176	100,0%	165	93,8%	168	95,5%	13 771	217	151
N	171	170	99,4%	151	88,3%	156	91,2%	13 313	291	151
O	150	148	98,7%	129	86,0%	137	91,3%	19 921	**538**	233
P	168	166	98,8%	144	85,7%	146	86,9%	18 594	229	208
Q	156	156	100,0%	146	93,6%	153	98,1%	21 415	317	244
R	140	139	99,3%	128	91,4%	134	95,7%	21 409	321	274
S	148	147	99,3%	128	86,5%	134	90,5%	13 764	357	165
T	167	166	99,4%	155	92,8%	157	94,0%	13 807	301	172
U	151	151	100,0%	139	92,1%	143	94,7%	18 245	327	254
V	154	152	98,7%	141	91,6%	146	94,8%	15 152	303	197
W	145	144	99,3%	140	96,6%	143	98,6%	17 223	247	208
X	155	155	100,0%	140	90,3%	148	95,5%	14 141	453	165
Y	155	153	98,7%	138	89,0%	138	89,0%	13 636	250	154
Z	146	143	97,9%	130	89,0%	136	93,2%	10 790	330	124
	4 000	**3 973**	**99,3%**	**3 556**	**88,9%**	**3 674**	**91,9%**	**402 720**	**538**	**308**

Appendix 8 – Statistics for T02, threshold 2

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	161	100,0%	152	94,4%	151	93,8%	16 988	227	193
B	154	153	99,4%	137	89,0%	141	91,6%	19 741	385	232
C	148	148	100,0%	133	89,9%	136	91,9%	16 698	250	191
D	164	162	98,8%	140	85,4%	149	90,9%	18 310	397	214
E	156	155	99,4%	136	87,2%	136	87,2%	14 102	361	164
F	158	157	99,4%	137	86,7%	147	93,0%	16 450	235	188
G	154	153	99,4%	138	89,6%	141	91,6%	21 968	343	256
H	145	141	97,2%	109	75,2%	115	79,3%	11 732	437	139
I	144	137	95,1%	126	87,5%	127	88,2%	5 373	249	64
J	162	160	98,8%	147	90,7%	153	94,4%	11 529	195	132
K	143	141	98,6%	122	85,3%	125	87,4%	15 819	290	183
L	145	145	100,0%	132	91,0%	136	93,8%	9 437	195	108
M	161	161	100,0%	152	94,4%	153	95,0%	14 251	216	161
N	163	161	98,8%	136	83,4%	142	87,1%	13 805	287	159
O	144	140	97,2%	126	87,5%	129	89,6%	19 995	**528**	237
P	151	149	98,7%	136	90,1%	137	90,7%	19 330	220	219
Q	139	139	100,0%	132	95,0%	134	96,4%	22 903	322	**265**
R	171	171	100,0%	160	93,6%	163	95,3%	20 183	289	233
S	136	134	98,5%	122	89,7%	125	91,9%	14 844	357	172
T	146	145	99,3%	126	86,3%	131	89,7%	14 266	300	163
U	180	180	100,0%	166	92,2%	164	91,1%	16 657	306	193
V	158	154	97,5%	137	86,7%	141	89,2%	15 944	282	183
W	142	142	100,0%	135	95,1%	138	97,2%	17 237	220	195
X	161	160	99,4%	154	95,7%	155	96,3%	13 990	450	162
Y	167	166	99,4%	153	91,6%	157	94,0%	13 975	239	158
Z	147	147	100,0%	133	90,5%	137	93,2%	11 211	319	130
	4 000	**3 962**	**99,1%**	**3 577**	**89,4%**	**3 663**	**91,6%**	**406 738**	**528**	**265**

Appendix 9 – Statistics for T03, threshold 2

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	160	99,4%	149	92,5%	155	96,3%	17 075	239	212
B	155	154	99,4%	135	87,1%	141	91,0%	19 867	414	250
C	149	148	99,3%	135	90,6%	140	94,0%	16 410	260	188
D	152	150	98,7%	139	91,4%	146	96,1%	18 290	438	215
E	149	149	100,0%	131	87,9%	134	89,9%	13 864	365	171
F	150	150	100,0%	134	89,3%	141	94,0%	16 551	250	202
G	151	151	100,0%	136	90,1%	140	92,7%	22 559	366	254
H	155	151	97,4%	114	73,5%	123	79,4%	11 143	440	138
I	163	161	98,8%	141	86,5%	147	90,2%	5 239	257	70
J	140	139	99,3%	128	91,4%	133	95,0%	11 705	194	144
K	144	142	98,6%	124	86,1%	133	92,4%	15 099	292	190
L	155	154	99,4%	139	89,7%	147	94,8%	9 282	202	98
M	143	143	100,0%	133	93,0%	139	97,2%	15 101	234	192
N	142	142	100,0%	121	85,2%	134	94,4%	14 157	319	171
O	170	169	99,4%	152	89,4%	159	93,5%	18 100	**527**	194
P	166	166	100,0%	138	83,1%	143	86,1%	18 519	227	222
Q	152	152	100,0%	136	89,5%	144	94,7%	21 987	331	**277**
R	166	166	100,0%	150	90,4%	156	94,0%	19 398	285	262
S	152	149	98,0%	122	80,3%	132	86,8%	14 835	341	198
T	166	163	98,2%	144	86,7%	154	92,8%	13 801	290	169
U	150	150	100,0%	132	88,0%	141	94,0%	17 659	310	201
V	141	139	98,6%	125	88,7%	130	92,2%	16 019	291	197
W	159	158	99,4%	154	96,9%	157	98,7%	16 944	217	190
X	153	153	100,0%	138	90,2%	140	91,5%	13 824	433	160
Y	151	151	100,0%	138	91,4%	146	96,7%	14 078	240	158
Z	165	164	99,4%	153	92,7%	152	92,1%	10 716	324	116
	4 000	**3974**	**99,4%**	**3 541**	**88,5%**	**3 707**	**92,7%**	**402 222**	**527**	**277**

Appendix 10 – Statistics for T04, threshold 2

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	152	100,0%	144	94,7%	145	95,4%	17 197	232	195
B	158	158	100,0%	144	91,1%	148	93,7%	19 779	395	253
C	151	151	100,0%	133	88,1%	137	90,7%	16 056	247	194
D	178	176	98,9%	162	91,0%	167	93,8%	17 460	409	200
E	168	167	99,4%	147	87,5%	154	91,7%	13 036	344	159
F	161	158	98,1%	129	80,1%	139	86,3%	16 070	250	196
G	150	150	100,0%	136	90,7%	142	94,7%	22 951	347	**258**
H	154	150	97,4%	117	76,0%	127	82,5%	11 090	434	139
I	157	156	99,4%	140	89,2%	147	93,6%	5 103	242	56
J	161	157	97,5%	143	88,8%	151	93,8%	11 160	177	123
K	131	130	99,2%	117	89,3%	119	90,8%	16 312	300	185
L	138	135	97,8%	111	80,4%	116	84,1%	9 560	190	116
M	162	161	99,4%	149	92,0%	153	94,4%	14 636	219	160
N	160	160	100,0%	136	85,0%	146	91,3%	13 774	297	160
O	140	139	99,3%	132	94,3%	134	95,7%	20 054	**558**	240
P	161	161	100,0%	147	91,3%	148	91,9%	19 016	234	217
Q	176	176	100,0%	162	92,0%	164	93,2%	21 532	314	244
R	129	129	100,0%	121	93,8%	122	94,6%	21 539	314	250
S	152	151	99,3%	129	84,9%	134	88,2%	14 158	358	165
T	144	143	99,3%	128	88,9%	134	93,1%	14 424	308	165
U	174	171	98,3%	146	83,9%	152	87,4%	17 573	335	204
V	146	144	98,6%	132	90,4%	136	93,2%	15 563	293	179
W	149	149	100,0%	141	94,6%	142	95,3%	17 086	231	196
X	155	154	99,4%	144	92,9%	141	91,0%	13 818	450	163
Y	159	159	100,0%	134	84,3%	146	91,8%	13 790	233	158
Z	134	132	98,5%	124	92,5%	127	94,8%	11 632	311	135
	4 000	**3 969**	**99,2%**	**3 548**	**88,7%**	**3 671**	**91,8%**	**404 369**	**558**	**258**

Appendix 11 – Statistics for T05, threshold 2

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	163	162	99,4%	151	92,6%	156	95,7%	16 144	239	181
B	150	149	99,3%	131	87,3%	136	90,7%	20 231	409	240
C	148	147	99,3%	132	89,2%	135	91,2%	16 566	264	189
D	143	143	100,0%	131	91,6%	133	93,0%	19 262	431	226
E	146	144	98,6%	121	82,9%	126	86,3%	14 348	372	169
F	174	172	98,9%	139	79,9%	151	86,8%	15 623	242	181
G	158	158	100,0%	141	89,2%	146	92,4%	20 955	358	246
H	132	131	99,2%	103	78,0%	106	80,3%	11 999	452	142
I	149	142	95,3%	133	89,3%	134	89,9%	5 479	254	65
J	143	143	100,0%	128	89,5%	132	92,3%	11 382	192	132
K	161	159	98,8%	136	84,5%	147	91,3%	15 053	286	174
L	146	144	98,6%	127	87,0%	128	87,7%	9 586	197	108
M	150	148	98,7%	132	88,0%	139	92,7%	14 521	229	167
N	147	146	99,3%	138	93,9%	138	93,9%	13 756	303	158
O	149	149	100,0%	137	91,9%	141	94,6%	19 920	**521**	234
P	157	155	98,7%	142	90,4%	143	91,1%	19 706	225	223
Q	160	159	99,4%	149	93,1%	154	96,3%	21 569	309	**248**
R	152	151	99,3%	131	86,2%	138	90,8%	20 613	300	237
S	160	159	99,4%	133	83,1%	144	90,0%	13 626	357	158
T	173	173	100,0%	156	90,2%	162	93,6%	13 628	295	154
U	158	156	98,7%	143	90,5%	145	91,8%	17 205	332	198
V	165	162	98,2%	153	92,7%	159	96,4%	14 940	291	170
W	157	157	100,0%	151	96,2%	156	99,4%	16 695	210	187
X	163	161	98,8%	141	86,5%	154	94,5%	14 037	469	160
Y	154	153	99,4%	138	89,6%	142	92,2%	13 851	250	157
Z	142	141	99,3%	122	85,9%	125	88,0%	11 164	333	127
	4 000	**3 964**	**99,1%**	**3 539**	**88,5%**	**3 670**	**91,8%**	**401 859**	**521**	**248**

Appendix 12 – Statistics for T01, threshold 3

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	151	99,3%	145	95,4%	146	96,1%	13 784	150	157
B	149	146	98,0%	127	85,2%	132	88,6%	14 305	239	166
C	140	138	98,6%	125	89,3%	127	90,7%	12 471	156	140
D	168	163	97,0%	138	82,1%	144	85,7%	12 505	255	146
E	149	149	100,0%	135	90,6%	140	94,0%	9 698	211	113
F	132	132	100,0%	115	87,1%	120	90,9%	12 270	141	143
G	160	155	96,9%	145	90,6%	143	89,4%	16 744	200	**195**
H	148	139	93,9%	114	77,0%	117	79,1%	7 302	259	86
I	142	137	96,5%	121	85,2%	122	85,9%	3 836	172	45
J	141	135	95,7%	118	83,7%	123	87,2%	9 218	103	104
K	160	158	98,8%	135	84,4%	142	88,8%	9 904	154	114
L	177	171	96,6%	144	81,4%	152	85,9%	6 129	107	68
M	176	175	99,4%	165	93,8%	166	94,3%	9 897	134	111
N	171	166	97,1%	150	87,7%	154	90,1%	9 409	187	108
O	150	147	98,0%	126	84,0%	133	88,7%	14 587	**345**	172
P	168	165	98,2%	140	83,3%	147	87,5%	13 663	125	164
Q	156	155	99,4%	144	92,3%	151	96,8%	15 799	193	185
R	140	138	98,6%	125	89,3%	131	93,6%	15 654	187	181
S	148	143	96,6%	124	83,8%	132	89,2%	9 213	200	106
T	167	165	98,8%	151	90,4%	155	92,8%	9 976	184	116
U	151	151	100,0%	138	91,4%	139	92,1%	13 759	197	160
V	154	151	98,1%	139	90,3%	145	94,2%	11 550	182	132
W	145	144	99,3%	139	95,9%	142	97,9%	12 826	133	146
X	155	154	99,4%	137	88,4%	146	94,2%	10 042	294	118
Y	155	151	97,4%	135	87,1%	134	86,5%	9 729	150	111
Z	146	142	97,3%	127	87,0%	134	91,8%	7 894	207	91
	4 000	**3 921**	**98,0%**	**3 502**	**87,6%**	**3 617**	**90,4%**	**292 164**	**345**	**195**

Appendix 13 – Statistics for T02, threshold 3

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	159	98,8%	149	92,5%	151	93,8%	13 697	151	156
B	154	149	96,8%	130	84,4%	135	87,7%	14 533	238	160
C	148	146	98,6%	133	89,9%	136	91,9%	12 489	151	133
D	164	162	98,8%	135	82,3%	146	89,0%	12 812	249	139
E	156	151	96,8%	131	84,0%	133	85,3%	9 768	215	106
F	158	155	98,1%	134	84,8%	147	93,0%	11 810	138	124
G	154	152	98,7%	136	88,3%	139	90,3%	16 436	210	177
H	145	135	93,1%	104	71,7%	114	78,6%	7 440	281	82
I	144	133	92,4%	124	86,1%	126	87,5%	3 930	174	43
J	162	157	96,9%	145	89,5%	152	93,8%	8 704	101	92
K	143	138	96,5%	115	80,4%	121	84,6%	10 813	168	121
L	145	144	99,3%	132	91,0%	136	93,8%	6 861	112	82
M	161	160	99,4%	150	93,2%	153	95,0%	10 140	135	117
N	163	159	97,5%	131	80,4%	140	85,9%	9 980	186	116
O	144	137	95,1%	126	87,5%	130	90,3%	14 379	**350**	176
P	151	148	98,0%	133	88,1%	135	89,4%	13 955	133	169
Q	139	139	100,0%	130	93,5%	133	95,7%	16 756	205	**193**
R	171	170	99,4%	159	93,0%	162	94,7%	14 294	165	163
S	136	134	98,5%	121	89,0%	126	92,6%	9 932	210	114
T	146	143	97,9%	125	85,6%	132	90,4%	10 502	203	120
U	180	175	97,2%	164	91,1%	163	90,6%	12 366	189	144
V	158	152	96,2%	134	84,8%	137	86,7%	12 283	181	141
W	142	142	100,0%	131	92,3%	135	95,1%	13 022	146	150
X	161	159	98,8%	150	93,2%	153	95,0%	9 727	286	115
Y	167	164	98,2%	151	90,4%	155	92,8%	9 964	149	115
Z	147	142	96,6%	130	88,4%	132	89,8%	8 212	221	96
	4 000	**3 905**	**97,6%**	**3 503**	**87,6%**	**3 622**	**90,6%**	**294 805**	**350**	**193**

Appendix 14 – Statistics for T03, threshold 3

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	157	97,5%	146	90,7%	148	91,9%	13 877	156	161
B	155	151	97,4%	131	84,5%	139	89,7%	14 458	246	176
C	149	146	98,0%	134	89,9%	138	92,6%	12 287	153	144
D	152	150	98,7%	135	88,8%	146	96,1%	12 717	263	152
E	149	146	98,0%	126	84,6%	132	88,6%	9 784	219	116
F	150	146	97,3%	131	87,3%	139	92,7%	11 907	142	138
G	151	150	99,3%	134	88,7%	132	87,4%	16 916	224	**197**
H	155	144	92,9%	108	69,7%	116	74,8%	7 015	272	84
I	163	155	95,1%	140	85,9%	146	89,6%	3 978	179	46
J	140	138	98,6%	124	88,6%	132	94,3%	8 743	116	103
K	144	140	97,2%	121	84,0%	127	88,2%	10 175	170	123
L	155	151	97,4%	137	88,4%	146	94,2%	6 633	123	77
M	143	141	98,6%	130	90,9%	137	95,8%	11 077	146	129
N	142	139	97,9%	119	83,8%	128	90,1%	10 151	195	123
O	170	167	98,2%	149	87,6%	160	94,1%	13 306	**336**	167
P	166	164	98,8%	137	82,5%	144	86,7%	13 512	127	156
Q	152	152	100,0%	137	90,1%	143	94,1%	16 318	219	190
R	166	164	98,8%	148	89,2%	152	91,6%	13 932	196	164
S	152	147	96,7%	117	77,0%	125	82,2%	10 093	231	123
T	166	161	97,0%	143	86,1%	149	89,8%	10 230	206	133
U	150	148	98,7%	134	89,3%	137	91,3%	13 489	233	157
V	141	136	96,5%	124	87,9%	128	90,8%	12 459	200	143
W	159	158	99,4%	152	95,6%	157	98,7%	12 664	141	143
X	153	151	98,7%	135	88,2%	138	90,2%	9 559	308	113
Y	151	151	100,0%	137	90,7%	144	95,4%	10 110	151	114
Z	165	163	98,8%	146	88,5%	152	92,1%	7 729	213	89
	4 000	**3 916**	**97,9%**	**3 475**	**86,9%**	**3 635**	**90,9%**	**293 119**	**336**	**197**

Appendix 15 – Statistics for T04, threshold 3

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	151	99,3%	141	92,8%	141	92,8%	13 905	143	159
B	158	158	100,0%	138	87,3%	149	94,3%	14 221	221	169
C	151	148	98,0%	134	88,7%	139	92,1%	11 976	142	136
D	178	175	98,3%	158	88,8%	165	92,7%	12 034	238	141
E	168	165	98,2%	143	85,1%	151	89,9%	8 875	198	104
F	161	154	95,7%	124	77,0%	139	86,3%	11 756	132	135
G	150	149	99,3%	131	87,3%	141	94,0%	16 896	199	**198**
H	154	145	94,2%	113	73,4%	125	81,2%	6 959	256	83
I	157	149	94,9%	139	88,5%	142	90,4%	3 698	169	41
J	161	153	95,0%	139	86,3%	147	91,3%	8 482	101	93
K	131	128	97,7%	109	83,2%	117	89,3%	11 022	162	128
L	138	132	95,7%	109	79,0%	114	82,6%	7 086	111	81
M	162	160	98,8%	146	90,1%	152	93,8%	10 581	131	120
N	160	160	100,0%	136	85,0%	144	90,0%	9 975	180	117
O	140	137	97,9%	131	93,6%	133	95,0%	14 539	**325**	174
P	161	159	98,8%	143	88,8%	147	91,3%	13 814	126	155
Q	176	175	99,4%	154	87,5%	164	93,2%	15 873	177	188
R	129	129	100,0%	121	93,8%	121	93,8%	15 805	172	182
S	152	149	98,0%	127	83,6%	133	87,5%	9 600	190	111
T	144	143	99,3%	124	86,1%	127	88,2%	10 783	180	128
U	174	168	96,6%	145	83,3%	152	87,4%	13 496	188	167
V	146	141	96,6%	128	87,7%	136	93,2%	12 009	179	138
W	149	147	98,7%	141	94,6%	142	95,3%	12 731	140	144
X	155	153	98,7%	141	91,0%	140	90,3%	9 582	286	123
Y	159	158	99,4%	130	81,8%	147	92,5%	9 868	144	113
Z	134	131	97,8%	118	88,1%	126	94,0%	8 557	208	101
	4 000	**3 917**	**97,9%**	**3 463**	**86,6%**	**3 634**	**90,9%**	**294 123**	**325**	**198**

Appendix 16 – Statistics for T05, threshold 3

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	163	159	97,5%	151	92,6%	153	93,9%	13 121	144	137
B	150	146	97,3%	123	82,0%	127	84,7%	14 802	236	179
C	148	144	97,3%	132	89,2%	137	92,6%	12 390	152	143
D	143	141	98,6%	129	90,2%	133	93,0%	13 504	259	159
E	146	141	96,6%	117	80,1%	125	85,6%	10 149	214	119
F	174	167	96,0%	138	79,3%	151	86,8%	11 414	136	131
G	158	156	98,7%	141	89,2%	144	91,1%	15 377	202	179
H	132	126	95,5%	100	75,8%	101	76,5%	7 647	266	91
I	149	142	95,3%	133	89,3%	135	90,6%	4 031	178	46
J	143	141	98,6%	122	85,3%	128	89,5%	8 657	105	99
K	161	156	96,9%	131	81,4%	141	87,6%	10 218	162	119
L	146	139	95,2%	125	85,6%	127	87,0%	6 953	117	80
M	150	146	97,3%	128	85,3%	136	90,7%	10 532	142	119
N	147	146	99,3%	136	92,5%	137	93,2%	9 877	189	120
O	149	147	98,7%	133	89,3%	137	91,9%	14 429	**326**	175
P	157	152	96,8%	141	89,8%	144	91,7%	14 318	129	173
Q	160	159	99,4%	148	92,5%	152	95,0%	15 790	188	**193**
R	152	148	97,4%	130	85,5%	135	88,8%	15 070	169	182
S	160	156	97,5%	126	78,8%	138	86,3%	9 092	195	108
T	173	171	98,8%	154	89,0%	158	91,3%	10 042	174	114
U	158	156	98,7%	138	87,3%	143	90,5%	13 026	196	162
V	165	162	98,2%	151	91,5%	158	95,8%	11 478	177	138
W	157	156	99,4%	150	95,5%	155	98,7%	12 411	140	141
X	163	160	98,2%	138	84,7%	149	91,4%	9 814	290	120
Y	154	152	98,7%	134	87,0%	142	92,2%	9 929	146	131
Z	142	139	97,9%	119	83,8%	124	87,3%	8 351	209	116
	4 000	**3 908**	**97,7%**	**3 468**	**86,7%**	**3 610**	**90,3%**	**292 422**	**326**	**193**

Appendix 17 – Statistics for T01, threshold 4

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	149	98,0%	143	94,1%	143	94,1%	10 610	104	**140**
B	149	143	96,0%	126	84,6%	130	87,2%	9 676	160	124
C	140	136	97,1%	123	87,9%	128	91,4%	8 743	102	94
D	168	153	91,1%	128	76,2%	137	81,5%	8 538	177	94
E	149	146	98,0%	132	88,6%	137	91,9%	6 520	146	71
F	132	128	97,0%	114	86,4%	118	89,4%	8 727	95	94
G	160	152	95,0%	137	85,6%	140	87,5%	11 785	129	127
H	148	126	85,1%	111	75,0%	115	77,7%	4 501	179	50
I	142	134	94,4%	120	84,5%	123	86,6%	2 968	134	31
J	141	130	92,2%	117	83,0%	121	85,8%	6 683	73	71
K	160	156	97,5%	132	82,5%	139	86,9%	6 353	105	69
L	177	164	92,7%	142	80,2%	145	81,9%	4 326	73	45
M	176	172	97,7%	157	89,2%	163	92,6%	6 888	93	73
N	171	158	92,4%	145	84,8%	148	86,5%	6 562	129	72
O	150	144	96,0%	122	81,3%	131	87,3%	10 113	**227**	124
P	168	153	91,1%	135	80,4%	143	85,1%	9 258	85	107
Q	156	155	99,4%	143	91,7%	148	94,9%	10 645	123	125
R	140	133	95,0%	122	87,1%	128	91,4%	10 802	115	129
S	148	138	93,2%	122	82,4%	127	85,8%	5 814	126	69
T	167	164	98,2%	148	88,6%	149	89,2%	6 892	122	80
U	151	151	100,0%	137	90,7%	138	91,4%	9 975	131	119
V	154	148	96,1%	138	89,6%	144	93,5%	8 502	126	100
W	145	143	98,6%	138	95,2%	141	97,2%	9 065	98	106
X	155	152	98,1%	134	86,5%	144	92,9%	6 836	205	83
Y	155	148	95,5%	131	84,5%	134	86,5%	6 650	101	78
Z	146	141	96,6%	126	86,3%	132	90,4%	5 589	153	63
	4 000	**3 817**	**95,4%**	**3 423**	**85,6%**	**3 546**	**88,7%**	**203 021**	**227**	**140**

Appendix 18 – Statistics for T02, threshold 4

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	157	97,5%	147	91,3%	150	93,2%	10 576	106	123
B	154	141	91,6%	126	81,8%	131	85,1%	10 286	163	126
C	148	143	96,6%	130	87,8%	137	92,6%	8 670	107	103
D	164	155	94,5%	129	78,7%	142	86,6%	8 693	183	103
E	156	146	93,6%	130	83,3%	134	85,9%	6 548	150	76
F	158	152	96,2%	131	82,9%	141	89,2%	8 315	96	98
G	154	151	98,1%	131	85,1%	135	87,7%	11 684	141	**138**
H	145	125	86,2%	106	73,1%	113	77,9%	4 697	193	55
I	144	132	91,7%	124	86,1%	126	87,5%	3 089	144	36
J	162	155	95,7%	143	88,3%	150	92,6%	6 380	75	73
K	143	132	92,3%	107	74,8%	114	79,7%	7 121	117	88
L	145	139	95,9%	130	89,7%	131	90,3%	4 742	83	53
M	161	157	97,5%	146	90,7%	151	93,8%	7 021	96	79
N	163	156	95,7%	130	79,8%	135	82,8%	7 128	135	83
O	144	132	91,7%	124	86,1%	125	86,8%	9 906	**249**	118
P	151	142	94,0%	130	86,1%	132	87,4%	9 436	86	108
Q	139	139	100,0%	127	91,4%	130	93,5%	11 311	133	131
R	171	168	98,2%	152	88,9%	163	95,3%	9 554	109	112
S	136	133	97,8%	118	86,8%	125	91,9%	6 211	139	73
T	146	136	93,2%	121	82,9%	129	88,4%	7 470	134	86
U	180	170	94,4%	159	88,3%	164	91,1%	8 802	133	103
V	158	148	93,7%	133	84,2%	137	86,7%	9 168	130	105
W	142	140	98,6%	130	91,5%	133	93,7%	9 160	105	104
X	161	158	98,1%	149	92,5%	152	94,4%	6 495	209	87
Y	167	161	96,4%	147	88,0%	152	91,0%	6 848	101	81
Z	147	138	93,9%	125	85,0%	127	86,4%	5 797	152	77
	4 000	**3 806**	**95,2%**	**3 425**	**85,6%**	**3 559**	**89,0%**	**205 108**	**249**	**138**

Appendix 19 – Statistics for T03, threshold 4

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	155	96,3%	144	89,4%	149	92,5%	10 749	100	133
B	155	145	93,5%	126	81,3%	132	85,2%	9 837	145	124
C	149	146	98,0%	130	87,2%	136	91,3%	8 704	96	112
D	152	149	98,0%	134	88,2%	143	94,1%	8 666	165	105
E	149	143	96,0%	119	79,9%	128	85,9%	6 557	137	88
F	150	142	94,7%	126	84,0%	136	90,7%	8 351	88	107
G	151	146	96,7%	130	86,1%	134	88,7%	12 074	133	**156**
H	155	135	87,1%	107	69,0%	114	73,5%	4 449	192	64
I	163	148	90,8%	137	84,0%	140	85,9%	3 174	135	45
J	140	134	95,7%	124	88,6%	130	92,9%	6 357	75	81
K	144	136	94,4%	118	81,9%	122	84,7%	6 706	102	87
L	155	149	96,1%	134	86,5%	142	91,6%	4 583	75	52
M	143	139	97,2%	127	88,8%	134	93,7%	7 862	104	90
N	142	138	97,2%	119	83,8%	127	89,4%	7 309	131	93
O	170	163	95,9%	146	85,9%	156	91,8%	9 289	**213**	121
P	166	155	93,4%	136	81,9%	138	83,1%	9 236	80	106
Q	152	150	98,7%	135	88,8%	140	92,1%	11 158	123	145
R	166	158	95,2%	138	83,1%	146	88,0%	9 425	106	110
S	152	138	90,8%	113	74,3%	121	79,6%	6 629	132	75
T	166	157	94,6%	139	83,7%	150	90,4%	7 181	121	99
U	150	144	96,0%	126	84,0%	134	89,3%	9 908	133	135
V	141	136	96,5%	120	85,1%	129	91,5%	9 316	125	118
W	159	158	99,4%	149	93,7%	155	97,5%	8 961	93	100
X	153	149	97,4%	135	88,2%	138	90,2%	6 521	208	90
Y	151	149	98,7%	132	87,4%	141	93,4%	6 991	98	93
Z	165	159	96,4%	146	88,5%	149	90,3%	5 475	141	62
	4 000	**3 821**	**95,5%**	**3 390**	**84,8%**	**3 564**	**89,1%**	**205 468**	**213**	**156**

Appendix 20 – Statistics for T04, threshold 4

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	148	97,4%	137	90,1%	141	92,8%	10 780	105	135
B	158	155	98,1%	135	85,4%	142	89,9%	9 651	151	122
C	151	147	97,4%	126	83,4%	140	92,7%	8 493	99	118
D	178	172	96,6%	155	87,1%	161	90,4%	8 053	166	114
E	168	160	95,2%	140	83,3%	150	89,3%	5 983	142	71
F	161	150	93,2%	122	75,8%	130	80,7%	8 437	92	97
G	150	148	98,7%	129	86,0%	138	92,0%	11 774	141	**147**
H	154	131	85,1%	109	70,8%	120	77,9%	4 321	189	52
I	157	147	93,6%	137	87,3%	142	90,4%	2 922	139	34
J	161	150	93,2%	137	85,1%	146	90,7%	6 249	75	77
K	131	124	94,7%	107	81,7%	113	86,3%	7 325	113	98
L	138	127	92,0%	109	79,0%	114	82,6%	5 028	89	65
M	162	158	97,5%	142	87,7%	148	91,4%	7 535	102	104
N	160	154	96,3%	132	82,5%	140	87,5%	7 108	142	84
O	140	134	95,7%	125	89,3%	131	93,6%	9 984	**255**	124
P	161	158	98,1%	141	87,6%	145	90,1%	9 492	91	119
Q	176	173	98,3%	152	86,4%	161	91,5%	10 713	125	133
R	129	126	97,7%	118	91,5%	121	93,8%	10 946	122	134
S	152	144	94,7%	126	82,9%	133	87,5%	6 235	135	77
T	144	136	94,4%	125	86,8%	126	87,5%	7 674	130	105
U	174	162	93,1%	141	81,0%	146	83,9%	9 935	143	117
V	146	139	95,2%	123	84,2%	133	91,1%	8 831	147	101
W	149	146	98,0%	141	94,6%	143	96,0%	8 924	107	104
X	155	151	97,4%	137	88,4%	136	87,7%	6 587	243	78
Y	159	153	96,2%	122	76,7%	143	89,9%	6 753	101	78
Z	134	128	95,5%	116	86,6%	121	90,3%	6 093	158	71
	4 000	**3 821**	**95,5%**	**3 384**	**84,6%**	**3 564**	**89,1%**	**205 826**	**255**	**147**

Appendix 21 – Statistics for T05, threshold 4

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	163	159	97,5%	150	92,0%	153	93,9%	10 042	103	116
B	150	136	90,7%	116	77,3%	125	83,3%	10 409	158	126
C	148	143	96,6%	127	85,8%	135	91,2%	8 840	101	102
D	143	139	97,2%	126	88,1%	130	90,9%	9 145	172	109
E	146	138	94,5%	114	78,1%	123	84,2%	6 980	148	83
F	174	163	93,7%	136	78,2%	146	83,9%	8 002	91	93
G	158	153	96,8%	137	86,7%	142	89,9%	10 829	138	128
H	132	118	89,4%	98	74,2%	104	78,8%	4 838	188	69
I	149	140	94,0%	132	88,6%	135	90,6%	3 050	136	36
J	143	136	95,1%	121	84,6%	124	86,7%	6 373	74	73
K	161	149	92,5%	123	76,4%	134	83,2%	6 739	104	79
L	146	136	93,2%	125	85,6%	126	86,3%	4 891	81	56
M	150	142	94,7%	128	85,3%	133	88,7%	7 391	101	96
N	147	145	98,6%	132	89,8%	136	92,5%	7 019	135	98
O	149	143	96,0%	126	84,6%	130	87,2%	9 939	**229**	**143**
P	157	147	93,6%	140	89,2%	141	89,8%	9 750	91	106
Q	160	158	98,8%	145	90,6%	151	94,4%	10 619	137	116
R	152	144	94,7%	129	84,9%	135	88,8%	10 408	120	114
S	160	148	92,5%	122	76,3%	134	83,8%	5 794	133	65
T	173	165	95,4%	152	87,9%	157	90,8%	6 983	123	76
U	158	151	95,6%	133	84,2%	138	87,3%	9 479	142	106
V	165	160	97,0%	151	91,5%	154	93,3%	8 528	129	94
W	157	155	98,7%	147	93,6%	151	96,2%	8 811	103	97
X	163	156	95,7%	136	83,4%	146	89,6%	6 628	224	82
Y	154	147	95,5%	131	85,1%	140	90,9%	6 981	112	86
Z	142	134	94,4%	119	83,8%	123	86,6%	6 146	166	79
	4 000	**3 805**	**95,1%**	**3 396**	**84,9%**	**3 546**	**88,7%**	**204 614**	**229**	**143**

Appendix 22 – Statistics for T01, threshold 5

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	147	96,7%	142	93,4%	145	95,4%	8 140	87	93
B	149	136	91,3%	121	81,2%	128	85,9%	6 758	125	94
C	140	133	95,0%	123	87,9%	126	90,0%	6 280	81	81
D	168	138	82,1%	111	66,1%	124	73,8%	6 208	139	75
E	149	142	95,3%	124	83,2%	134	89,9%	4 531	117	54
F	132	126	95,5%	114	86,4%	119	90,2%	6 435	75	75
G	160	145	90,6%	129	80,6%	139	86,9%	8 457	104	**100**
H	148	122	82,4%	108	73,0%	112	75,7%	2 981	149	36
I	142	130	91,5%	115	81,0%	123	86,6%	2 469	116	29
J	141	126	89,4%	114	80,9%	117	83,0%	4 956	60	57
K	160	146	91,3%	127	79,4%	133	83,1%	4 329	80	52
L	177	152	85,9%	137	77,4%	144	81,4%	3 140	58	36
M	176	165	93,8%	153	86,9%	158	89,8%	4 956	77	57
N	171	155	90,6%	145	84,8%	149	87,1%	4 722	105	56
O	150	141	94,0%	122	81,3%	128	85,3%	7 205	**187**	88
P	168	151	89,9%	131	78,0%	137	81,5%	6 471	66	75
Q	156	151	96,8%	140	89,7%	147	94,2%	7 270	99	86
R	140	130	92,9%	115	82,1%	124	88,6%	7 634	90	92
S	148	131	88,5%	116	78,4%	123	83,1%	3 781	99	54
T	167	157	94,0%	147	88,0%	150	89,8%	4 879	96	60
U	151	148	98,0%	133	88,1%	136	90,1%	7 314	109	86
V	154	145	94,2%	134	87,0%	141	91,6%	6 363	96	74
W	145	143	98,6%	135	93,1%	141	97,2%	6 460	78	74
X	155	149	96,1%	129	83,2%	140	90,3%	4 821	169	58
Y	155	143	92,3%	128	82,6%	134	86,5%	4 701	79	54
Z	146	138	94,5%	122	83,6%	129	88,4%	4 064	124	48
	4 000	**3 690**	**92,3%**	**3 315**	**82,9%**	**3 481**	**87,0%**	**145 325**	**187**	**100**

Appendix 23 – Statistics for T02, threshold 5

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	154	95,7%	147	91,3%	149	92,5%	8 219	82	97
B	154	133	86,4%	120	77,9%	128	83,1%	7 434	119	**109**
C	148	141	95,3%	129	87,2%	134	90,5%	6 136	77	71
D	164	147	89,6%	125	76,2%	134	81,7%	6 325	133	73
E	156	142	91,0%	128	82,1%	133	85,3%	4 530	106	54
F	158	147	93,0%	129	81,6%	138	87,3%	6 043	70	71
G	154	144	93,5%	126	81,8%	132	85,7%	8 567	106	102
H	145	119	82,1%	104	71,7%	113	77,9%	3 177	145	38
I	144	127	88,2%	121	84,0%	125	86,8%	2 586	110	30
J	162	152	93,8%	141	87,0%	149	92,0%	4 708	51	55
K	143	125	87,4%	106	74,1%	111	77,6%	5 044	78	60
L	145	136	93,8%	127	87,6%	133	91,7%	3 435	58	39
M	161	150	93,2%	142	88,2%	145	90,1%	4 978	77	57
N	163	146	89,6%	130	79,8%	131	80,4%	5 181	98	62
O	144	130	90,3%	121	84,0%	126	87,5%	7 105	**187**	87
P	151	137	90,7%	127	84,1%	133	88,1%	6 477	65	75
Q	139	134	96,4%	124	89,2%	129	92,8%	7 642	105	90
R	171	163	95,3%	147	86,0%	157	91,8%	6 538	77	78
S	136	129	94,9%	116	85,3%	124	91,2%	3 973	101	46
T	146	133	91,1%	118	80,8%	126	86,3%	5 462	95	64
U	180	168	93,3%	158	87,8%	161	89,4%	6 333	99	76
V	158	147	93,0%	128	81,0%	136	86,1%	6 863	102	76
W	142	138	97,2%	129	90,8%	133	93,7%	6 607	79	77
X	161	158	98,1%	145	90,1%	151	93,8%	4 553	172	55
Y	167	157	94,0%	142	85,0%	144	86,2%	4 947	74	57
Z	147	134	91,2%	123	83,7%	124	84,4%	4 125	122	48
	4 000	**3 691**	**92,3%**	**3 353**	**83,8%**	**3 499**	**87,5%**	**146 988**	**187**	**109**

Appendix 24 – Statistics for T03, threshold 5

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	161	152	94,4%	144	89,4%	147	91,3%	8 413	86	98
B	155	134	86,5%	120	77,4%	128	82,6%	6 848	119	84
C	149	142	95,3%	127	85,2%	136	91,3%	6 257	82	73
D	152	145	95,4%	132	86,8%	138	90,8%	6 149	140	75
E	149	134	89,9%	118	79,2%	125	83,9%	4 669	120	56
F	150	141	94,0%	125	83,3%	134	89,3%	6 100	75	71
G	151	140	92,7%	125	82,8%	129	85,4%	8 764	112	**105**
H	155	124	80,0%	105	67,7%	110	71,0%	3 025	151	36
I	163	146	89,6%	139	85,3%	141	86,5%	2 699	121	31
J	140	131	93,6%	121	86,4%	127	90,7%	4 711	58	54
K	144	130	90,3%	115	79,9%	122	84,7%	4 730	80	56
L	155	145	93,5%	131	84,5%	138	89,0%	3 291	64	37
M	143	137	95,8%	129	90,2%	134	93,7%	5 707	84	71
N	142	133	93,7%	120	84,5%	125	88,0%	5 376	112	63
O	170	161	94,7%	143	84,1%	152	89,4%	6 841	**192**	84
P	166	151	91,0%	134	80,7%	140	84,3%	6 570	65	77
Q	152	147	96,7%	128	84,2%	136	89,5%	7 549	102	89
R	166	149	89,8%	136	81,9%	144	86,7%	6 632	86	79
S	152	129	84,9%	111	73,0%	119	78,3%	4 429	103	53
T	166	154	92,8%	133	80,1%	148	89,2%	5 200	102	60
U	150	138	92,0%	123	82,0%	127	84,7%	7 387	111	90
V	141	136	96,5%	120	85,1%	128	90,8%	7 103	103	83
W	159	158	99,4%	147	92,5%	155	97,5%	6 450	78	74
X	153	149	97,4%	131	85,6%	138	90,2%	4 638	182	56
Y	151	148	98,0%	128	84,8%	139	92,1%	5 097	82	60
Z	165	155	93,9%	143	86,7%	146	88,5%	3 950	123	46
	4 000	**3 709**	**92,7%**	**3 328**	**83,2%**	**3 506**	**87,7%**	**148 585**	**192**	**105**

Appendix 25 – Statistics for T04, threshold 5

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	152	146	96,1%	138	90,8%	140	92,1%	8 293	85	99
B	158	150	94,9%	135	85,4%	141	89,2%	6 670	116	84
C	151	143	94,7%	124	82,1%	137	90,7%	6 093	79	75
D	178	170	95,5%	152	85,4%	158	88,8%	5 687	130	70
E	168	153	91,1%	135	80,4%	148	88,1%	4 213	112	50
F	161	143	88,8%	117	72,7%	129	80,1%	6 152	74	73
G	150	140	93,3%	124	82,7%	131	87,3%	8 475	110	**103**
H	154	127	82,5%	100	64,9%	116	75,3%	2 997	148	36
I	157	144	91,7%	133	84,7%	137	87,3%	2 437	120	29
J	161	146	90,7%	132	82,0%	143	88,8%	4 671	56	54
K	131	120	91,6%	106	80,9%	108	82,4%	5 060	86	60
L	138	118	85,5%	102	73,9%	111	80,4%	3 699	65	42
M	162	155	95,7%	136	84,0%	143	88,3%	5 405	79	63
N	160	147	91,9%	131	81,9%	136	85,0%	5 168	110	61
O	140	132	94,3%	122	87,1%	128	91,4%	7 032	**192**	87
P	161	151	93,8%	141	87,6%	145	90,1%	6 634	67	78
Q	176	168	95,5%	144	81,8%	155	88,1%	7 197	100	95
R	129	123	95,3%	115	89,1%	120	93,0%	7 743	89	92
S	152	139	91,4%	120	78,9%	126	82,9%	4 147	101	49
T	144	130	90,3%	121	84,0%	122	84,7%	5 477	101	66
U	174	158	90,8%	139	79,9%	145	83,3%	7 376	105	88
V	146	135	92,5%	119	81,5%	127	87,0%	6 640	105	95
W	149	145	97,3%	141	94,6%	143	96,0%	6 272	78	82
X	155	146	94,2%	134	86,5%	137	88,4%	4 714	183	57
Y	159	147	92,5%	118	74,2%	135	84,9%	4 845	78	56
Z	134	126	94,0%	110	82,1%	120	89,6%	4 474	127	58
	4 000	**3 702**	**92,6%**	**3 289**	**82,2%**	**3 481**	**87,0%**	**147 571**	**192**	**103**

Appendix 26 – Statistics for T05, threshold 5

letter	TL	MA1	AC1	MA2	AC2	MA3	AC3	R	RT	CT
A	163	155	95,1%	150	92,0%	152	93,3%	7 781	77	91
B	150	126	84,0%	112	74,7%	116	77,3%	7 414	115	87
C	148	142	95,9%	129	87,2%	133	89,9%	6 411	75	94
D	143	134	93,7%	122	85,3%	123	86,0%	6 335	128	80
E	146	136	93,2%	109	74,7%	123	84,2%	5 024	112	60
F	174	159	91,4%	135	77,6%	140	80,5%	5 796	67	78
G	158	147	93,0%	130	82,3%	135	85,4%	7 859	101	96
H	132	114	86,4%	95	72,0%	99	75,0%	3 374	146	48
I	149	138	92,6%	131	87,9%	134	89,9%	2 555	111	30
J	143	129	90,2%	119	83,2%	121	84,6%	4 830	55	56
K	161	141	87,6%	121	75,2%	131	81,4%	4 699	78	57
L	146	130	89,0%	122	83,6%	126	86,3%	3 519	62	49
M	150	139	92,7%	123	82,0%	129	86,0%	5 326	78	75
N	147	142	96,6%	128	87,1%	133	90,5%	5 091	104	66
O	149	138	92,6%	123	82,6%	129	86,6%	7 049	**176**	95
P	157	144	91,7%	137	87,3%	141	89,8%	6 827	65	89
Q	160	156	97,5%	141	88,1%	147	91,9%	7 241	96	**105**
R	152	136	89,5%	124	81,6%	129	84,9%	7 343	80	104
S	160	140	87,5%	117	73,1%	128	80,0%	3 795	98	45
T	173	163	94,2%	149	86,1%	155	89,6%	4 912	92	58
U	158	148	93,7%	131	82,9%	139	88,0%	7 118	101	88
V	165	159	96,4%	147	89,1%	153	92,7%	6 505	99	95
W	157	155	98,7%	142	90,4%	149	94,9%	6 341	77	89
X	163	155	95,1%	133	81,6%	143	87,7%	4 727	160	60
Y	154	141	91,6%	125	81,2%	129	83,8%	5 052	78	73
Z	142	132	93,0%	115	81,0%	123	86,6%	4 552	115	53
	4 000	**3 699**	**92,5%**	**3 310**	**82,8%**	**3 460**	**86,5%**	**147 476**	**176**	**105**

Appendix 27 – Statistics for T01-T05, threshold 1, average

letter	AC1	AC2	AC3	R	RT	CT
A	99,9%	94,2%	95,7%	17 975	612	206
B	99,6%	88,1%	91,2%	21 379	1 113	249
C	99,7%	89,5%	91,8%	17 605	774	199
D	99,6%	89,4%	92,1%	19 946	1 171	240
E	99,5%	87,2%	90,3%	15 353	1 011	184
F	99,4%	85,3%	90,9%	17 739	750	205
G	99,7%	90,8%	93,2%	23 844	1 025	**278**
H	98,9%	76,9%	81,0%	13 041	1 167	150
I	98,0%	88,9%	90,0%	6 003	578	70
J	99,0%	89,4%	92,5%	12 442	578	140
K	99,3%	86,7%	90,2%	17 107	922	199
L	99,1%	86,7%	90,5%	10 186	582	119
M	99,6%	92,8%	94,9%	15 524	589	178
N	99,7%	87,8%	91,9%	15 023	796	177
O	99,3%	90,6%	93,5%	21 082	**1 272**	254
P	99,2%	88,4%	89,6%	20 297	676	230
Q	100,0%	93,0%	95,8%	23 219	876	267
R	99,7%	91,5%	94,3%	22 135	943	256
S	98,9%	86,3%	89,6%	15 748	1 040	180
T	99,2%	88,9%	92,8%	15 230	819	174
U	99,5%	90,3%	91,3%	18 880	828	218
V	98,3%	91,4%	92,9%	16 770	744	194
W	99,9%	96,3%	97,8%	18 306	580	205
X	99,6%	91,2%	93,6%	15 402	1 131	179
Y	99,9%	89,6%	92,7%	15 126	697	170
Z	99,6%	91,0%	92,4%	12 101	794	142
	99,4%	**89,3%**	**92,0%**	**437 463**	**1 272**	**278**

Appendix 28 – Statistics for T01-T05, threshold 2, average

letter	AC1	AC2	AC3	R	RT	CT
A	99,8%	93,7%	95,6%	16 922	237	201
B	99,6%	88,6%	91,4%	19 902	400	250
C	99,6%	89,5%	91,7%	16 490	258	194
D	99,1%	89,0%	92,4%	18 260	422	217
E	99,5%	87,2%	89,8%	13 876	361	168
F	99,3%	84,9%	90,7%	16 317	246	200
G	99,6%	90,2%	92,5%	22 175	357	**264**
H	98,1%	76,4%	80,4%	11 522	440	142
I	97,6%	87,7%	90,3%	5 286	252	63
J	98,7%	89,0%	92,3%	11 572	189	134
K	99,0%	86,0%	90,6%	15 445	290	181
L	98,9%	86,0%	89,8%	9 268	193	105
M	99,6%	92,2%	95,0%	14 456	223	166
N	99,5%	87,2%	91,6%	13 761	300	160
O	98,9%	89,8%	93,0%	19 598	**534**	228
P	99,2%	88,1%	89,4%	19 033	227	218
Q	99,9%	92,6%	95,7%	21 881	319	256
R	99,7%	91,1%	94,1%	20 628	302	251
S	98,9%	84,9%	89,5%	14 245	354	172
T	99,2%	89,0%	92,6%	13 985	299	164
U	99,4%	89,3%	91,8%	17 468	322	210
V	98,3%	90,0%	93,2%	15 524	292	185
W	99,7%	95,9%	97,8%	17 037	225	195
X	99,5%	91,1%	93,7%	13 962	451	162
Y	99,5%	89,2%	92,8%	13 866	242	157
Z	99,0%	90,1%	92,3%	11 103	324	126
	99,2%	**88,8%**	**91,9%**	**403 582**	**534**	**264**

Appendix 29 – Statistics for T01-T05, threshold 3, average

letter	AC1	AC2	AC3	R	RT	CT
A	98,5%	92,8%	93,7%	13 677	149	154
B	97,9%	84,7%	89,0%	14 464	236	170
C	98,1%	89,4%	92,0%	12 323	151	139
D	98,3%	86,4%	91,3%	12 714	253	147
E	97,9%	84,9%	88,7%	9 655	212	112
F	97,4%	83,1%	89,9%	11 831	138	134
G	98,6%	88,9%	90,4%	16 474	207	189
H	93,9%	73,5%	78,0%	7 273	267	85
I	94,8%	87,0%	88,8%	3 895	174	44
J	97,0%	86,7%	91,2%	8 761	105	98
K	97,4%	82,7%	87,7%	10 426	163	121
L	96,8%	85,1%	88,7%	6 732	114	77
M	98,7%	90,7%	93,9%	10 445	138	119
N	98,4%	85,9%	89,9%	9 878	187	117
O	97,6%	88,4%	92,0%	14 248	**337**	173
P	98,1%	86,5%	89,3%	13 852	128	163
Q	99,6%	91,2%	94,9%	16 107	196	**190**
R	98,8%	90,1%	92,5%	14 951	178	175
S	97,5%	82,4%	87,6%	9 586	205	113
T	98,4%	87,5%	90,5%	10 307	189	122
U	98,2%	88,5%	90,4%	13 227	200	158
V	97,1%	88,4%	92,1%	11 956	183	138
W	99,3%	94,8%	97,2%	12 731	140	145
X	98,7%	89,1%	92,2%	9 745	293	118
Y	98,7%	87,4%	91,9%	9 920	148	117
Z	97,7%	87,2%	91,0%	8 149	212	99
	97,8%	**87,1%**	**90,6%**	**293 327**	**337**	**190**

Appendix 30 – Statistics for T01-T05, threshold 4, average

letter	AC1	AC2	AC3	R	RT	CT
A	97,3%	91,4%	93,3%	10 551	103	129
B	94,0%	82,1%	86,1%	9 972	155	125
C	97,1%	86,4%	91,8%	8 690	101	106
D	95,5%	83,6%	88,7%	8 619	173	105
E	95,5%	82,6%	87,5%	6 518	145	78
F	94,9%	81,4%	86,8%	8 366	92	98
G	97,0%	85,9%	89,2%	11 629	136	**139**
H	86,6%	72,4%	77,2%	4 561	188	58
I	92,9%	86,1%	88,2%	3 041	137	37
J	94,4%	85,9%	89,7%	6 408	74	75
K	94,3%	79,5%	84,2%	6 849	108	84
L	94,0%	84,2%	86,6%	4 714	80	54
M	96,9%	88,3%	92,0%	7 339	99	88
N	96,0%	84,1%	87,8%	7 025	134	86
O	95,0%	85,4%	89,3%	9 846	**235**	126
P	94,1%	85,0%	87,1%	9 434	86	109
Q	99,0%	89,8%	93,3%	10 889	128	130
R	96,2%	87,1%	91,5%	10 227	114	120
S	93,8%	80,5%	85,7%	6 137	133	72
T	95,2%	86,0%	89,2%	7 240	126	89
U	95,8%	85,7%	88,6%	9 620	136	116
V	95,7%	86,9%	91,2%	8 869	132	104
W	98,7%	93,7%	96,1%	8 984	101	102
X	97,3%	87,8%	91,0%	6 613	218	84
Y	96,4%	84,4%	90,3%	6 845	103	83
Z	95,3%	86,0%	88,8%	5 820	154	71
	95,4%	**85,1%**	**88,9%**	**204 807**	**235**	**139**

Appendix 31 – Statistics for T01-T05, threshold 5, average

letter	AC1	AC2	AC3	R	RT	CT
A	95,6%	91,4%	92,9%	8 169	84	96
B	88,6%	79,3%	83,6%	7 025	119	92
C	95,2%	85,9%	90,5%	6 235	79	79
D	91,3%	80,0%	84,2%	6 141	134	75
E	92,1%	79,9%	86,3%	4 593	114	55
F	92,5%	80,3%	85,5%	6 105	72	73
G	92,6%	82,0%	86,2%	8 424	107	**101**
H	82,7%	69,9%	75,0%	3 111	148	39
I	90,7%	84,6%	87,4%	2 549	116	30
J	91,5%	83,9%	87,8%	4 775	56	55
K	89,6%	77,9%	81,9%	4 772	80	57
L	89,6%	81,4%	85,8%	3 417	62	40
M	94,2%	86,3%	89,6%	5 274	79	65
N	92,5%	83,6%	86,2%	5 108	106	62
O	93,2%	83,8%	88,1%	7 046	**187**	88
P	91,4%	83,5%	86,8%	6 596	65	79
Q	96,6%	86,6%	91,3%	7 380	100	93
R	92,6%	84,2%	89,0%	7 178	84	89
S	89,4%	77,8%	83,1%	4 025	100	49
T	92,5%	83,8%	87,9%	5 186	97	61
U	93,6%	84,1%	87,1%	7 106	105	86
V	94,5%	84,7%	89,6%	6 695	101	85
W	98,2%	92,3%	95,9%	6 426	78	79
X	96,2%	85,4%	90,1%	4 691	173	57
Y	93,7%	81,6%	86,7%	4 928	78	60
Z	93,3%	83,4%	87,5%	4 233	122	51
	92,5%	83,0%	87,1%	**147 189**	**187**	**101**

Appendix 32 – “Nursery” data set

class	**K01a**	**K02a**	**K03a**	**K04a**	**K05a**	**Total**
not_recom	899	879	837	845	860	**4 320**
priority	867	849	873	865	812	**4 266**
spec_prior	760	798	816	816	854	**4 044**
	2 526	**2 526**	**2 526**	**2 526**	**2 526**	**12 630**

Appendix 33 – Statistics for T01a-T05a, thresholds 1

letter	**TL**	**MA1**	**AC1**	**MA2**	**AC2**	**MA3**	**AC3**	**R**	**RT**	**CT**
not_recom	860	860	100,0%	859	99,9%	860	100,0%	167	11,5	2,5
priority	812	810	99,8%	722	88,9%	740	91,1%	728	14,5	3,9
spec_prior	854	853	99,9%	739	86,5%	780	91,3%	633	13,3	3,8
T01a	**2 526**	**2 523**	**99,9%**	**2320**	**91,8%**	**2 380**	**94,2%**	**1 528**	**14,5**	**3,9**
not_recom	899	899	100,0%	899	100,0%	899	100,0%	153	10,9	2,4
priority	867	862	99,4%	774	89,3%	800	92,3%	661	14,3	3,6
spec_prior	760	760	100,0%	676	88,9%	708	93,2%	682	13,6	3,7
T02a	**2 526**	**2 521**	**99,8%**	**2349**	**93,0%**	**2 407**	**95,3%**	**1 496**	**14,3**	**3,7**
not_recom	879	879	100,0%	879	100,0%	879	100,0%	160	11,0	2,4
priority	849	849	100,0%	772	90,9%	787	92,7%	670	14,1	3,6
spec_prior	798	796	99,7%	723	90,6%	754	94,5%	688	13,4	3,7
T03a	**2 526**	**2 524**	**99,9%**	**2374**	**94,0%**	**2 420**	**95,8%**	**1 518**	**14,1**	**3,7**
not_recom	837	837	100,0%	837	100,0%	837	100,0%	159	11,0	2,4
priority	873	869	99,5%	780	89,3%	805	92,2%	684	14,9	3,7
spec_prior	816	813	99,6%	710	87,0%	761	93,3%	687	13,3	5,0
T04a	**2 526**	**2 519**	**99,7%**	**2 327**	**92,1%**	**2 403**	**95,1%**	**1 530**	**14,9**	**5,0**
not_recom	845	845	100,0%	845	100,0%	845	100,0%	173	10,9	3,3
priority	865	865	100,0%	748	86,5%	776	89,7%	691	14,3	3,7
spec_prior	816	813	99,6%	709	86,9%	758	92,9%	680	13,4	3,7
T05a	**2 526**	**2 523**	**99,9%**	**2 302**	**91,1%**	**2 379**	**94,2%**	**1 544**	**14,3**	**3,7**
not_recom	864		100,0%		100,0%		100,0%	162	11,1	2,6
priority	853		99,7%		89,0%		91,6%	687	14,4	3,7
spec_prior	809		99,8%		88,0%		93,0%	674	13,4	4,0
AVG	**2 526**		**99,8%**		**92,4%**		**94,9%**	**1 523**	**14,4**	**4,0**

Appendix 34 – Statistics for T01a-T05a, thresholds 2

letter	**TL**	**MA1**	**AC1**	**MA2**	**AC2**	**MA3**	**AC3**	**R**	**RT**	**CT**
not_recom	860	860	100,0%	860	100,0%	860	100,0%	3	10,9	1,7
priority	812	812	100,0%	800	98,5%	803	98,9%	576	13,0	3,2
spec_prior	854	854	100,0%	773	90,5%	807	94,5%	527	13,5	3,1
T01a	**2 526**	**2 526**	**100,0%**	**2 433**	**96,3%**	**2 470**	**97,8%**	**1 106**	**13,5**	**3,2**
not_recom	899	899	100,0%	899	100,0%	899	100,0%	8	10,8	2,0
priority	867	862	99,4%	849	97,9%	850	98,0%	512	13,2	3,2
spec_prior	760	759	99,9%	702	92,4%	729	95,9%	563	13,0	3,3
T02a	**2 526**	**2 520**	**99,8%**	**2 450**	**97,0%**	**2 478**	**98,1%**	**1 083**	**13,2**	**3,3**
not_recom	879	879	100,0%	879	100,0%	879	100,0%	9	10,6	1,7
priority	849	848	99,9%	836	98,5%	839	98,8%	542	13,1	3,1
spec_prior	798	796	99,7%	734	92,0%	769	96,4%	572	13,6	3,2
T03a	**2 526**	**2 523**	**99,9%**	**2 449**	**97,0%**	**2 487**	**98,5%**	**1 123**	**13,6**	**3,2**
not_recom	837	837	100,0%	837	100,0%	837	100,0%	10	10,5	1,7
priority	873	869	99,5%	858	98,3%	862	98,7%	534	13,1	3,1
spec_prior	816	809	99,1%	727	89,1%	779	95,5%	568	13,3	3,3
T04a	**2 526**	**2 515**	**99,6%**	**2 422**	**95,9%**	**2 478**	**98,1%**	**1 112**	**13,3**	**3,3**
not_recom	845	845	100,0%	845	100,0%	845	100,0%	10	10,7	1,7
priority	865	864	99,9%	853	98,6%	856	99,0%	541	13,3	3,1
spec_prior	816	811	99,4%	740	90,7%	787	96,4%	553	12,7	3,1
T05a	**2 526**	**2 520**	**99,8%**	**2 438**	**96,5%**	**2 488**	**98,5%**	**1 104**	**13,3**	**3,1**
not_recom	864		100,0%		100,0%		100,0%	8	10,7	1,8
priority	853		99,7%		98,4%		98,7%	541	13,1	3,1
spec_prior	809		99,6%		90,9%		95,7%	557	13,2	3,2
AVG	**2 526**		**99,8%**		**96,5%**		**98,2%**	**1 106**	**13,4**	**3,2**

Appendix 35 – Statistics for T01a-T05a, thresholds 3

letter	**TL**	**MA1**	**AC1**	**MA2**	**AC2**	**MA3**	**AC3**	**R**	**RT**	**CT**
not_recom	860	860	100,0%	860	100,0%	860	100,0%	2	11,2	1,9
priority	812	808	99,5%	800	98,5%	803	98,9%	304	12,8	2,1
spec_prior	854	852	99,8%	778	91,1%	812	95,1%	305	12,8	2,1
T01a	**2 526**	**2 520**	**99,8%**	**2 438**	**96,5%**	**2 475**	**98,0%**	**611**	**12,8**	**2,1**
not_recom	899	899	100,0%	899	100,0%	899	100,0%	2	10,8	1,9
priority	867	857	98,8%	855	98,6%	857	98,8%	282	12,4	2,0
spec_prior	760	759	99,9%	705	92,8%	732	96,3%	343	12,1	2,3
T02a	**2 526**	**2 515**	**99,6%**	**2 459**	**97,3%**	**2 488**	**98,5%**	**627**	**12,4**	**2,3**
not_recom	879	879	100,0%	879	100,0%	879	100,0%	3	10,3	1,7
priority	849	847	99,8%	842	99,2%	844	99,4%	306	12,5	2,0
spec_prior	798	796	99,7%	735	92,1%	772	96,7%	354	13,0	2,3
T03a	**2 526**	**2 522**	**99,8%**	**2 456**	**97,2%**	**2 495**	**98,8%**	**663**	**13,0**	**2,3**
not_recom	837	837	100,0%	837	100,0%	837	100,0%	3	10,4	1,6
priority	873	869	99,5%	867	99,3%	867	99,3%	304	12,6	2,0
spec_prior	816	809	99,1%	733	89,8%	783	96,0%	365	12,0	2,3
T04a	**2 526**	**2 515**	**99,6%**	**2 437**	**96,5%**	**2 487**	**98,5%**	**672**	**12,6**	**2,3**
not_recom	845	845	100,0%	845	100,0%	845	100,0%	1	10,6	1,6
priority	865	860	99,4%	857	99,1%	857	99,1%	302	12,2	2,0
spec_prior	816	811	99,4%	745	91,3%	792	97,1%	323	12,7	2,1
T05a	**2 526**	**2 516**	**99,6%**	**2 447**	**96,9%**	**2 494**	**98,7%**	**626**	**12,7**	**2,1**
not_recom	864		100,0%		100,0%		100,0%	2	10,7	1,7
priority	853		99,4%		98,9%		99,1%	300	12,5	2,0
spec_prior	809		99,6%		91,4%		96,2%	338	12,5	2,2
AVG	**2 526**		**99,7%**		**96,9%**		**98,5%**	**640**	**12,7**	**2,2**

Appendix 36 – Statistics for T01a-T05a, thresholds 4

letter	**TL**	**MA1**	**AC1**	**MA2**	**AC2**	**MA3**	**AC3**	**R**	**RT**	**CT**
not_recom	860	860	100,0%	860	100,0%	860	100,0%	1	9,9	1,7
priority	812	808	99,5%	805	99,1%	807	99,4%	259	11,9	1,9
spec_prior	854	852	99,8%	787	92,2%	819	95,9%	274	12,4	2,0
T01a	**2 526**	**2 520**	**99,8%**	**2 452**	**97,1%**	**2 486**	**98,4%**	**534**	**12,4**	**2,0**
not_recom	899	899	100,0%	899	100,0%	899	100,0%	1	9,7	1,7
priority	867	855	98,6%	855	98,6%	855	98,6%	229	12,0	1,7
spec_prior	760	759	99,9%	715	94,1%	736	96,8%	303	12,1	2,1
T02a	**2 526**	**2 513**	**99,5%**	**2 469**	**97,7%**	**2 490**	**98,6%**	**533**	**12,1**	**2,1**
not_recom	879	879	100,0%	879	100,0%	879	100,0%	1	9,6	1,7
priority	849	846	99,6%	844	99,4%	846	99,6%	257	11,7	1,8
spec_prior	798	796	99,7%	741	92,9%	777	97,4%	306	12,0	2,1
T03a	**2 526**	**2 521**	**99,8%**	**2 464**	**97,5%**	**2 502**	**99,0%**	**564**	**12,0**	**2,1**
not_recom	837	837	100,0%	837	100,0%	837	100,0%	2	9,6	1,6
priority	873	861	98,6%	859	98,4%	859	98,4%	242	11,9	1,8
spec_prior	816	809	99,1%	742	90,9%	792	97,1%	323	12,0	2,1
T04a	**2 526**	**2 507**	**99,2%**	**2 438**	**96,5%**	**2 488**	**98,5%**	**567**	**12,0**	**2,1**
not_recom	845	845	100,0%	845	100,0%	845	100,0%	1	9,7	1,6
priority	865	860	99,4%	858	99,2%	858	99,2%	258	12,2	1,9
spec_prior	816	811	99,4%	753	92,3%	797	97,7%	291	11,5	2,0
T05a	**2 526**	**2 516**	**99,6%**	**2 456**	**97,2%**	**2500**	**99,0%**	**550**	**12,2**	**2,0**
not_recom	864		100,0%		100,0%		100,0%	1	9,7	1,7
priority	853		99,2%		98,9%		99,0%	249	11,9	1,8
spec_prior	809		99,6%		92,4%		97,0%	299	12,0	2,1
AVG	**2 526**		**99,6%**		**97,2%**		**98,7%**	**550**	**12,1**	**2,1**

Appendix 37 – Statistics for T01a-T05a, thresholds 5

letter	**TL**	**MA1**	**AC1**	**MA2**	**AC2**	**MA3**	**AC3**	**R**	**RT**	**CT**
not_recom	860	860	100,0%	860	100,0%	860	100,0%	1	9,7	1,7
priority	812	808	99,5%	806	99,3%	808	99,5%	203	11,6	1,6
spec_prior	854	852	99,8%	793	92,9%	823	96,4%	243	11,0	1,8
T01a	**2 526**	**2 520**	**99,8%**	**2 459**	**97,3%**	**2 491**	**98,6%**	**447**	**11,6**	**1,8**
not_recom	899	899	100,0%	899	100,0%	899	100,0%	1	9,6	1,7
priority	867	850	98,0%	850	98,0%	850	98,0%	185	11,1	1,4
spec_prior	760	759	99,9%	721	94,9%	738	97,1%	270	11,1	1,9
T02a	**2 526**	**2 508**	**99,3%**	**2 470**	**97,8%**	**2 487**	**98,5%**	**456**	**11,1**	**1,9**
not_recom	879	879	100,0%	879	100,0%	879	100,0%	1	9,4	1,7
priority	849	841	99,1%	839	98,8%	841	99,1%	200	11,2	1,5
spec_prior	798	796	99,7%	749	93,9%	780	97,7%	265	11,8	1,9
T03a	**2 526**	**2 516**	**99,6%**	**2 467**	**97,7%**	**2 500**	**99,0%**	**466**	**11,8**	**1,9**
not_recom	837	837	100,0%	837	100,0%	837	100,0%	1	9,6	1,6
priority	873	859	98,4%	859	98,4%	859	98,4%	202	11,1	1,5
spec_prior	816	806	98,8%	745	91,3%	790	96,8%	276	11,4	1,9
T04a	**2 526**	**2 502**	**99,0%**	**2 441**	**96,6%**	**2 486**	**98,4%**	**479**	**11,4**	**1,9**
not_recom	845	845	100,0%	845	100,0%	845	100,0%	1	9,3	1,6
priority	865	856	99,0%	858	99,2%	858	99,2%	209	11,1	1,6
spec_prior	816	809	99,1%	753	92,3%	797	97,7%	259	11,0	1,8
T05a	**2 526**	**2 510**	**99,4%**	**2 456**	**97,2%**	**2 500**	**99,0%**	**469**	**11,1**	**1,8**
not_recom	864		100,0%		100,0%		100,0%	1	9,5	1,7
priority	853		98,8%		98,7%		98,8%	200	11,2	1,5
spec_prior	809		99,5%		93,0%		97,1%	263	11,3	1,9
AVG	**2 526**		**99,4%**		**97,3%**		**98,7%**	**463**	**11,4**	**1,9**