

Journal of Information Technology and Applications

(BANJA LUKA)

JITA

Exchange of Information
and Knowledge in Research

APEIRON
ЖУРНАЛ



VOLUME 3

NUMBER 2

BANJA LUKA, DECEMBER 2013 (61-116)

ISSN 2232-9625 (Print)

UDC 004

THE AIM AND SCOPE

The aim and scope of the Journal of Information Technology and Applications (JITA) is:

- to provide international dissemination of contributions in field of Information Technology,
- to promote exchange of information and knowledge in research work and
- to explore the new developments and inventions related to the use of Information Technology towards the structuring of an Information Society.

JITA provides a medium for exchanging research results and achievements accomplished by the scientific community from academia and industry.

By the decision of the Ministry of Education and Culture of the Republic of Srpska, no.: 07.030-053-160-4/10 from 3/3/2010, the journal „Journal of Information Technology and Applications“ Banja Luka is registered in the Registry of public organs under the number 591. Printed by Markos, Banja Luka in 300 copies two times a year.

Indexed in: LICENSE AGREEMENT, 3.22.12. EBSCO Publishing Inc., Current Abstracts

 EBSCOHOST.COM

 INDEX COPERNICUS INTERNATIONAL INDEXCOPERNICUS.COM

 Google SCHOLAR.GOOGLE.COM

 DOISRPSKA.NUB.RS

 CROSSREF.ORG

Printed on acid-free paper

Full-text available free of charge at <http://www.jita-au.com>

CONTENTS

COMPARATIVE ANALYSIS OF DATA MINING TECHNIQUES APPLIED TO WIRELESS SENSOR NETWORK DATA FOR FIRE DETECTION	65
<i>MIRJANA MAKSIMOVIĆ, VLADIMIR VUJOVIĆ</i> <i>Faculty of Electrical Engineering, University of East Sarajevo</i>	
ENUMERATION, RANKING AND GENERATION OF BINARY TREES BASED ON LEVEL-ORDER TRAVERSAL USING CATALAN CIPHER VECTORS.....	78
<i>ADRIJAN BOŽINOVSKI, BILJANA STOJČEVSKA, VENO PAČOVSKI</i> <i>University American College Skopje</i>	
USING DECISION TREE CLASSIFIER FOR ANALYZING STUDENTS' ACTIVITIES	87
<i>SNJEŽANA MILINKOVIĆ, MIRJANA MAKSIMOVIĆ</i> <i>Faculty of Electrical Engineering, East Sarajevo, Bosnia and Herzegovina</i>	
OBJECT-ORIENTED ANALYSIS AND DESIGN FOR ONE ALGORITHM OF COMPUTATIONAL GEOMETRY: FORWARD, REVERSE AND ROUND-TRIP ENGINEERING	96
<i>MUZAFER H. SARAČEVIĆ, PREDRAG S. STANIMIROVIĆ, SEAD H. MAŠOVIĆ</i> <i>Faculty of Science and Mathematics, University of Nis</i>	
CRM PERFORMANCES ACCENTED WITH THE IMPLEMENTATION OF DATA WAREHOUSING AND DATA MINING TECHNOLOGIES.....	107
<i>INES ISAKOVIĆ</i> <i>JU Zavod za prostorno uređenje Blhać</i>	
INSTRUCTIONS FOR AUTHORS	113

EDITORS:



**GORDANA
RADIĆ, PhD**
EDITOR-IN-CHIEF



**ZORAN
AVRAMOVIĆ, PhD**



**DUŠAN
STARČEVIĆ, PhD**

Dear Readers, we introduce you the 6th issue of the JITA Journal.

Out of the submitted papers, reviewers have selected five papers that will be presented in this issue.

The paper titled „*Comparative analysis of data Mining techniques applied to wireless sensor network data for fire detection*“ by Mirjana Maksimović and Vladimir Vujović focuses on a comparative analysis of various Data Mining techniques and algorithms. For that purpose, three experiments in the WSN fire detection were carried out. For fire detection, the most suitable classification by means of fuzzy logic is shown.

The paper „Using Decision Tree classifier for analyzing students activities“ by Snježana Milinković and Mirjana Maksimović focuses on the analysis of student data from the Moodle Database and data collected manually at the Faculty of Electrical Engineering in Istočno Sarajevo. Using the method of classification J48 decision tree, four experiments were carried out. Obtained results provide a significant contribution to the paper.

The paper titled „Object oriented analysis and design for one algorithm of computational geometry: forward reverse and round trip engineering“ by Muzafer Saračević, Predrag Stanimirović and Sead Mašović demonstrates key advantages of the object oriented analysis and design in solving convex polygon triangulations. Based on the presented testing, it has been concluded that the synchronization technique which combines Java programming and UML modelling is the most suitable one.

The paper „Enumeration, ranking and generation binary trees based on level-order traversal using Catalan Cipher Vectors“ by Adrijan Božinovski, Biljana Stojčevska and Venio Pančevski deals with the presentation and detailed analysis of new representation of a binary tree, Catalan Cipher Vectors.

The paper „CRM performances accentuated with the implementation of Data Warehousing and Data Mining technologies“ by Ines Isaković represents CRM along with DW and DM as an essential system of modern business organisation.

We thank the authors for the effort they have invested in order to present the results of their research in a high quality manner. We wish for our presented papers to be recognized both by the readers and scientific community.

COMPARATIVE ANALYSIS OF DATA MINING TECHNIQUES APPLIED TO WIRELESS SENSOR NETWORK DATA FOR FIRE DETECTION

Mirjana Maksimović¹, Vladimir Vujović²

¹mirjana@etf.unssa.rs.ba, ²vladimir.vujovic@etf.unssa.rs.ba

Contribution to the state of the art

DOI: 10.7251/JIT1302065M

UDC: 519.816:004.735

Abstract: Wireless sensor networks (WSN) are a rapidly growing area for research and commercial development with very wide range of applications. Using WSNs many critical events like fire can be detected earlier to prevent losing human lives and heavy structural damages. Integration of soft computing techniques on sensor nodes, like fuzzy logic, neural networks and data mining, can significantly lead to improvements of critical events detection possibility. Using data mining techniques in process of patterns discovery in large data sets it's not often so easy. A several algorithms must be applied to application before a suitable algorithm for selected data types can be found. Therefore, the selection of a correct data mining algorithm depends on not only the goal of an application, but also on the compatibility of the data set. This paper focuses on comparative analysis of various data mining techniques and algorithms and in that purpose three different experiments on WSN fire detection data are proposed and performed. The primary goal was to see which of them has the best classification accuracy of fuzzy logic generated data and is the most appropriate for a particular application of fire detection.

Keywords: Analysis, Data Mining, Fire Detection, WEKA, WSN

INTRODUCTION

With the advancement in sensors' technology sensor networks are increasingly finding its applications in many domains such as human activity monitoring [14], vehicle monitoring [8], vibration analysis [13], habitat monitoring [15], object tracking [3], environment monitoring [9, 10, 16] including critical events detections [1,17] etc. A critical event, like fire can cause heavy structural damage to the indoor area and life threatening conditions so early residential fire detection is important for prompt extinguishing and reducing damages and life losses. To detect fire, one or a combination of sensors and a detection algorithm are needed where the sensors might be part of a wireless sensor network (WSN) or work independently [1].

The extraction of useful knowledge from raw sensor data is a difficult task and traditional data min-

ing techniques are not directly applicable to WSNs due to the distributed nature of sensor data and their special characteristics (the massive quantity and the high dimensionality), and limitations of the WSNs and sensor nodes. This is the reason for exploring novel data mining techniques dealing with extracting knowledge from large continuous arriving data from WSNs [11]. For such reasons, in recent years a great interest emerged in the research community in applying data mining techniques to the large volumes of sensor data. Sensor data mining is a relatively new area but it already reached a certain level of maturity.

Data mining, as an iterative process of extracting hidden patterns from large data sets and a critical component of the knowledge discovery process, consists of a collection of automated and semi-automated techniques for modeling relationships and uncovering hidden patterns in large data repositories. It

draws upon ideas from diverse disciplines such as statistics, machine learning, pattern recognition, database systems, information theory, and artificial intelligence [18]. Sensor data brings numerous challenges with it in the context of data collection, storage and processing and variety of data mining methods such as clustering, classification, frequent pattern mining, and outlier detection are often applied to sensor data in order to extract actionable insights (Fig. 1).

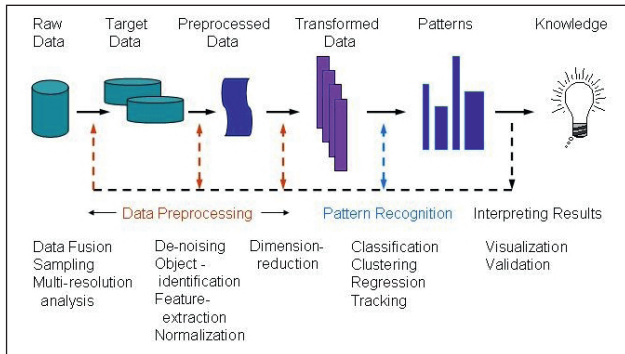


FIGURE 1 THE OVERALL PROCESS OF KNOWLEDGE DISCOVERY FROM DATA INCLUDES DATA PREPROCESSING, DATA MINING, AND POSTPROCESSING OF THE DATA MINING RESULTS

On the one hand, massive volumes of disparate data, typically dimensioned by space and time, are being generated in real time or near real time. On the other hand, the need for faster and more reliable decisions is growing rapidly in the face of emerging challenges like fire. One critical path to enhanced threat recognition is through online knowledge discovery based on dynamic, heterogeneous data available from strategically placed wide-area sensor networks. The knowledge discovery process needs to coordinate adaptive predictive analysis with real-time analysis and decision support systems. The ability to detect precursors and signatures of rare events and change from massive and disparate data in real time is a challenge [4].

The goal of predictive modeling is to build a model that can be used to predict - based on known examples collected in the past - future values of a target attribute. There are many predictive modeling methods available, including tree-based, rule-based, nearest neighbor, logistic regression, artificial neural networks, graphical methods, and support vector machines. These methods are designed to solve two

types of predictive modeling tasks: classification and regression [11]. Using these prediction models the number of sensors that need to report their measurements is reduced by reducing both node activity and bandwidth. From analysis made in [11] it is observed that the techniques intended for mining sensor data at network side are helpful for taking real time decision as well as serve as prerequisite for development of effective mechanism for data storage, retrieval, query and transaction processing at central side. On the other hand centralized techniques are helpful in generating off-line predictive insights which in turn can facilitate real-time analysis.

The massive streams of sensor data generated in some applications make it impossible to use algorithms that must store the entire data into main memory. Using data mining techniques in process of patterns discovery in large data sets it's not often so easy. A several algorithms must be applied to application before a suitable algorithm for selected data types can be found. Online algorithms provide an attractive alternative to conventional batch algorithms for handling such large data sets. The selection of a correct data mining algorithm depends on not only the goal of an application, but also on the compatibility of the data set. This paper focuses on comparative analysis of various data mining techniques and algorithms with primary goal to see which of them has the best classification accuracy and is the most appropriate for a particular application of fire detection uncovering useful information hidden in large quantities of sensor data. This kind of analysis provide an opportunity for data mining researchers to develop more advanced methods for handling some of the issues specific to sensor data.

The rest of this paper is organized as following. Second section presents data preparation file while third section provides an implementation of selected data mining techniques. The experimental results including comparative analysis of selected algorithms are shown in fourth section. Fifth section gives the conclusion.

FIRE DETECTION – PREPARING THE INPUT FILES

Early detection of critical events, like residential

fire, is crucial for life saving and reduction of potential damages so WSN should be able to detect if fire has occurred or is about to. But just like many other human-recognizable events, the phenomenon fire has no real meaning to a sensor node. Therefore, suitable techniques that would allow describing events in ways that sensor nodes would be able to “understand” are needed. One of them is fuzzy technique. What makes fuzzy logic suitable for use in WSNs is that it can tolerate unreliable and imprecise sensor readings, it is much closer to human way of thinking than crisp logic and compared to other classification algorithms based on probability theory, fuzzy logic is much more intuitive and easier to use. It allows using linguistic variables whose values are not numbers but words or sentences in a natural or artificial language. Fuzzy rules are conditional statements in the form of IF-THEN which:

- Require less computational power than conventional mathematical computational methods,
- Require few data samples in order to extract the final result,
- and the most important, it can be effectively manipulated since they use human language to describe problems (based on heuristic information that mainly comes from expert knowledge of the system) and making the creation of rules simple, independently of the previous knowledge in the field of fuzzy logic.

Preparing input for a data mining investigation usually consumes the bulk of the effort invested in the entire data mining process. However, simple application of data mining technique to sensor data may not be as successful as expected because sensor data are mostly mere numerical values. Thus, contextual data should be incorporated in the database for data mining as well as sensor data [22].

In this work three different experiments for fire detection will be presented based on similar approaches given in [2, 7, 19]. For the sake of clarity of machine learning domain the correlated sensor data used for a detection of fire are converted to nominal types [12]. Input data are defined as IF-THEN rules based on heuristic information that mainly comes from expert knowledge of the fire detection systems.

The massive streams of sensor data which could be generated in fire detection applications make it impossible to use algorithms that must store the entire data into main memory. For that purpose, on full rule-base consisted of fuzzy rules for detection of fire, presented in the rest of the paper, FURIA (Fuzzy Unordered Rule Induction Algorithm) will be applied. Other four chosen algorithms will be compared to results obtained using FURIA with aim to realize which of them generate the best prediction models uncovering useful information hidden in large quantities of sensor data in a case of fire detection.

The three proposed experiment were created with main goal to show how chosen algorithms predicting power depends on number of data and the fire detection method.

In first experiment, detection of fire is based on two heat detectors - fixed heat and rate of rise heat detector [7]. A fixed temperature heat detector utilizes a temperature sensing element which will generate an alarm condition if the temperature within the protected area reaches a predetermined level (e.g. 57 °C, 63 °C, 74 °C or 90 °C) while rate of rise heat detector is a device that responds when the temperature rises at a rate exceeding a predetermined value (e.g. 8.33 °C/min, 9 °C/min or 11 °C/min, according to NFPA 72 standard). Instead of using these crisp values, fuzzy logic proposes use of linguistic variables. Therefore, data obtained from those two temperature detectors according to fuzzy technique and above mentioned thresholds, for the purpose of the experiment are described with values: *very low* (VL), *low* (L), *medium* (M), *high* (H) and *very high* (VH) and presented with membership functions shown in Fig. 2 and Fig. 3, respectively. Due to their simple formulas and computational efficiency, both triangular and trapezoidal membership functions have been used extensively, especially in real-time implementations as it is fire detection.

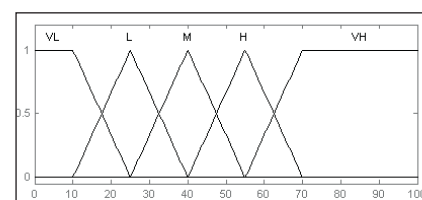


FIGURE 2 THE MEMBERSHIP FUNCTION OF INPUT VARIABLE

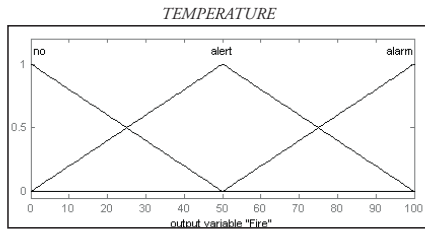


FIGURE 3 THE MEMBERSHIP FUNCTION OF INPUT VARIABLE TEMPERATURE DIFFERENCE

Possibility of *fire* is defined as output variable and is described with *no*, *alert* and *alarm* linguistic variables as it shown in Fig. 4. This linguistic variable represents the system’s confidence in the presence of fire. For example, if the fire confidence is smaller than 50, the probability that there is no fire is higher. If the fire confidence value is higher than 80, there is more than 80% possibility that there is a fire.

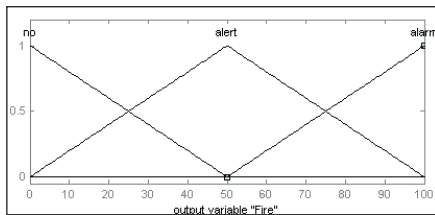


FIGURE 4 THE MEMBERSHIP FUNCTION OF OUTPUT VARIABLE FIRE

With 2 variables each of which can take 5 values, the number of rules in the full fuzzy rule-base of first experiment is 25 (5*5). Table 1 shows first 10 rules for 1st experiment.

TABLE 1 THE 1ST FIRE DATA TEST (FIRST 10 RULES)

Temperature difference	Temperature	Fire (class)
VL	VL	no
L	VL	no
M	VL	no
H	VL	alert
VH	VL	alarm
VL	L	no
L	L	no
M	L	alert
H	L	alert
VH	L	alarm

In the second experiment, detection of fire is based on two successively measured fixed heat temperature detector data [19] in function of additional variable time. Previous and current values of temperature are the same as in Fig. 2. Third input variable time is described with two linguistic variables: short (S) and

long (L), according to °C/min changes (Fig. 5).

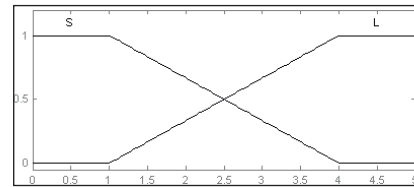


FIGURE 5 THE MEMBERSHIP FUNCTION OF INPUT VARIABLE TIME

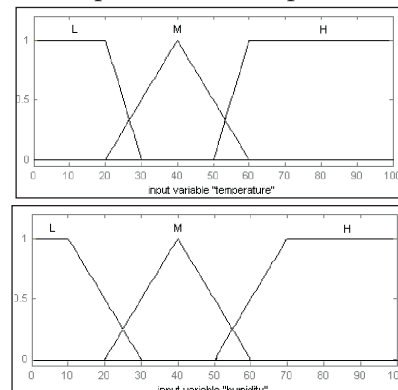
Output variable fire is the same as presented in Fig. 4.

In this case, there are 3 variables and the number of rules in the rule-base is 50 (5*5*2). Table 2 shows first 10 rules of the second experiment.

TABLE 2 THE 2ND FIRE DATA TEST (FIRST 10 RULES)

Previous temperature	Current temperature	time	Fire (class)
VL	VL	S	no
VL	VL	L	no
VL	L	S	no
VL	L	L	no
VL	M	S	alert
VL	M	L	no
VL	H	S	alert
VL	H	L	alert
VL	VH	S	alarm
VL	VH	L	alert

The third experiment considers that fire detection is not based only on the temperature values but also on the CO, humidity and light levels, similar as in [2]. Therefore, proposed fire detection logic in this case takes four linguistic variables as input – *temperature*, *humidity*, *light* and *CO*. The linguistic values for all four input variables are classified into *low* (L), *medium* (M), and *high* (H) (Fig. 6). Output variable *fire* is the same as in previous two experiments.



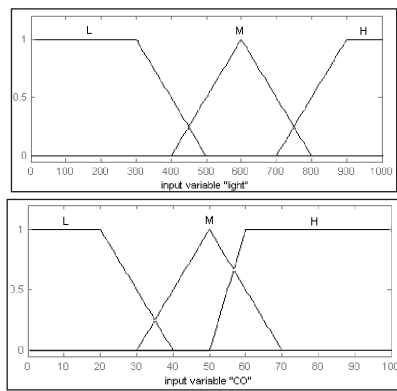


FIGURE 6 THE MEMBERSHIP FUNCTIONS OF INPUT VARIABLES TEMPERATURE, HUMIDITY, LIGHT AND CO

With 4 variables each of which can take 3 values, the number of rules in the rule-base is 81 (3*3*3*3). Table 3 shows first 10 rules for third fire detection scenario.

TABLE 3 THE 3RD FIRE DATA TEST (FIRST 10 RULES)

Temperature	Humidity	Light	CO	Fire (class)
L	L	L	L	no
L	L	L	M	alert
L	L	L	H	alert
L	L	M	L	no
L	L	M	M	alert
L	L	M	H	alarm
L	L	H	L	no
L	L	H	M	alert
L	L	H	H	alarm
L	M	L	L	no

For further analysis Excel .csv data files are formed based on data given in Tables 1, 2 and 3. The next step is their exporting to WEKA data mining tool [20] in order to apply chosen classification algorithms presented in the rest of the paper.

CLASSIFICATION ALGORITHMS IMPLEMENTATIONS

Implementations of chosen classification algorithms are performed in WEKA, which is a collection of machine learning algorithms for data mining tasks. The algorithms in WEKA can be applied directly to a previous formed data sets as it is used in this paper. The main advantage of using WEKA is to apply the learning methods to a data set and analyze its output to extract information about the data. These learning methods are called classifiers. In simulation process the classifiers from WEKA in

order to analyze the classification accuracy of simulation data are used. Classification here means the problem of correctly predicting the probability that an example has a predefined class from a set of attributes describing the example. The purpose is to apply the learning algorithms and then to choose the best one for prediction purposes [21].

There are many methods and measures for estimation the strength and the accuracy of a classification/predictive model. The main measure is the classification accuracy which is the number of correctly classified instances in the test set divided by the total number of instances in the test set. Some of the common methods for classifier evaluation are holdout set, Multiple Random Sampling and Cross-validation.

The output of the simulator proposed in this paper is used to learn the difference between a subject that is *no*, *alert* and *alarm*. For these experiments averaging and 10-fold cross validation testing techniques are used. During the process the data set is divided into 10 subsets. Then the classification algorithms are fed with these subsets of data. The left-out subsets of the training data are used to evaluate classification accuracy. When seeking an accurate error estimate, it is standard procedure to repeat the cross-validation process 10 times (that is 10 times tenfold cross-validation) and average the results. This involves invoking the learning algorithm 100 times on data sets that are all nine-tenths the size of the original. Getting a good measure of performance is a computation-intensive undertaking [21].

In applications with only two classes two measures named Precision and Recall are usually used. Their definitions are:

$$P = \frac{TP}{TP + FP} \quad (1) \quad R = \frac{TP}{TP + FN} \quad (2)$$

TP, FP and FN used in Eq. (1) and Eq. (2) are the numbers of true positives, false positives and false negatives, respectively. These measures can be also used in case of larger number of classes, which in this case are seen as a series of problems with two classes. It is convenient to introduce these measures using a

confusion matrix. A confusion matrix contains information about actual and predicted results given by a classifier. However, it is hard to compare classifiers based on two measures, which are not functionally related [21].

If a single measure to compare different classifiers is needed, the F-measure is often used:

$$FM = \frac{2 \cdot P \cdot R}{P + R} \quad (3)$$

Another measure is the receiver operating characteristic (ROC). It is a term used in signal detection to characterize the tradeoff between hit rate and false-alarm rate over a noisy channel. ROC curves depict the performance of a classifier without regard to class distribution or error costs. They plot the true positive rate on the vertical axis against the true negative rate on the horizontal axis.

In addition, it is possible to evaluate attributes by measuring their information gain with respect to the class using Info-Gain Attribute Evaluation and measuring their gain ratio with respect to the class using Gain-Ratio Attribute Evaluation [21]. Information gain is biased towards multivalued attributes while gain ratio tends to prefer unbalanced splits in which one partition is much smaller than the others.

In simulation process presented in this paper four widely used classification algorithms [21] are implemented for comparative analysis with FURIA on given fire data sets. Thus, the comparative analysis is based on following algorithms:

- FURIA
- OneR
- J48 decision tree
- Naive Bayes
- Neural Network classifier

FURIA

FURIA (Fuzzy Unordered Rule Induction Algorithm) is a fuzzy rule-based classification method proposed in 2009 by Hühn and Hüllermeier [6]. FURIA extends the well-known RIPPER algorithm preserving its advantages, such as simple and comprehensible rule sets. In addition, FURIA includes a

number of modifications and extensions. It obtains fuzzy rules instead of the usual strict rules, as well as an unordered rule set instead of the rule list. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method. The idea is to generalize the existing rules until they cover the example [6].

OneR

OneR is classifier with one parameter – the minimum bucket size for discretization. It generates a one-level decision tree expressed in the form of a set of rules that all test one particular attribute. OneR is a simple, cheap method that often comes up with quite good rules for characterizing the structure in data. In any event, it is always a good plan to try the simplest things first. The idea of OneR is to make rules that test a single attribute and branch accordingly. Next step is to use the class that occurs most often in the training data and to determine the error rate of the rules counting the errors that occur on the training data (the number of instances that do not have the majority class) [21].

Pseudocode of OneR algorithm is:

For each attribute,

For each value of that attribute, make a rule as follows:

count how often each class appears

find the most frequent class

make the rule assign that class to this attribute value.

Calculate the error rate of the rules.

Choose the rules with the smallest error rate.

Decision Tree Classifier

WEKA uses the J48 decision tree which is an implementation of the C 4.5 algorithm. The decision tree classifier is a tree based classifier which selects a set of features and then compares the input data with them and its main advantage is classification speed. Learned patterns are represented as a tree where nodes in the tree embody decisions based on the values of attributes and the leaves of the tree provide predictions [21].

Naïve Bayes

The Naïve Bayes classifier, for each class value, estimates the probability that a given instance belongs to that class. It is a statistical classifier and performs probabilistic prediction, i.e., predicts class membership probabilities. A simple Bayesian classifier, Naïve Bayes Classifier (based on Bayes' theorem.), has comparable performance with decision tree and selected neural network classifiers. Each training example can incrementally increase/decrease the probability that a hypothesis is correct - prior knowledge can be combined with observed data. Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision [5]. Naïve Bayes gives a simple approach, with clear semantics, for representing, using, and learning probabilistic knowledge and it can achieve impressive results [21].

Neural network classifier

The Neural network classifier is used for many pattern recognition purposes. It uses the backpropagation algorithm to train the network. The accuracy of the neural network classifiers does not depend on the dimensionality of the training data [21].

In rest of the paper comparative analysis, using FURIA as base predictive model, will be performed.

COMPARATIVE ANALYSIS OF SIMULATION RESULTS

Simulation results (performances and classifier error) of above described experiments and chosen algorithms are shown in rest of the paper. It will be shown which of applied algorithms has the highest percentage of correct classified instances (CCI), the minimal of incorrect classified instances (ICI), the highest precision (P) and the classification above ROC curve area in function of chosen experiment and its number of data.

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	60	40	0.6	0.314	0.51	0.6	0.506	0.704
OneR	40	60	0.4	0.297	0.4	0.4	0.395	0.552
J48	40	60	0.4	0.297	0.4	0.4	0.395	0.682
NB	48	52	0.48	0.266	0.436	0.48	0.455	0.661
NN	56	44	0.56	0.25	0.516	0.56	0.531	0.731

1st experiment

Attributes evaluation of data used in 1st experiment are shown in Table 4.

TABLE 4. ATTRIBUTES EVALUATION – 1ST EXPERIMENT

Attribute	InfoGainAttributeEval	GainRatioAttributeEval
Temperature	0.248	0.107
Temperature difference	0.602	0.259

Applying FURIA classifier to existing rules shown in Table 1, 25 rules are generalized into only 3 (Table 5).

TABLE 5 THE FIRE DATA TEST OBTAINED USING FURIA IN 1ST EXPERIMENT

Temperature difference	Temperature	Fire (class)
VL	/	no
VH	/	alarm
/	VH	alarm

J48 decision tree for presented fire data in 1st experiment is shown in Fig. 7. The attribute with the maximum gain ratio, as it is showed in Table 4, is *temperature difference* and it is selected as the splitting attribute.

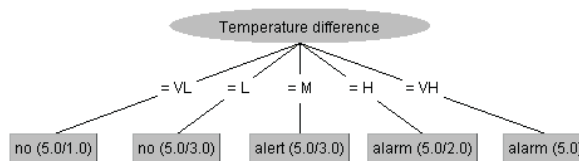


FIGURE 7 J48 DECISION TREE – 1ST EXPERIMENT

Classifiers evaluation is presented in Table 6.

TABLE 6 CLASSIFIER EVALUATION – 1ST EXPERIMENT

From Table 4 it can be seen that FURIA has the best prediction model. It generated a model with 60% correctly classified instances (CCI), a precision of 51% (0.51) and the classification above the ROC curve area (0.704 > 0.5).

In multiclass prediction, the result on a test set is often displayed as a two-dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements [21]. The results are shown in Table 7.

TABLE 7 CONFUSION MATRICES – 1ST EXPERIMENT

FURIA				OneR			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
4	0	3	<i>a=no</i>	5	2	0	<i>a=no</i>
0	0	7	<i>b=alert</i>	3	0	4	<i>b=alert</i>
0	0	11	<i>c=alarm</i>	1	5	5	<i>c=alarm</i>

J48				Naïve Bayes			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
5	2	0	<i>a=no</i>	5	2	0	<i>a=no</i>
3	0	4	<i>b=alert</i>	3	0	4	<i>b=alert</i>
1	5	5	<i>c=alarm</i>	1	3	7	<i>c=alarm</i>

Neural Network			
Predicted class			
a	b	c	Real class
5	2	0	<i>a=no</i>
1	1	5	<i>b=alert</i>
2	1	8	<i>c=alarm</i>

Applying Resample filter on data given in Table 1, the balance of data distribution is significantly improved what affect the results of the applied algorithms. In other words, it is possible to generate model with more precise predictions. Results obtained by applying above mentioned algorithms on re-sampled data are shown in next tables. Table 8 shows the predictive accuracy of the classifier on the re-sampled data. From Table 8 it can be seen that J48 decision tree and OneR classifiers have the best prediction models. On re-sampled data they generated a model with 80% correctly classified instances (CCI) and precision of 82.8% (0.828).

Confusion matrices of re-sampled data are presented in Table 9.

TABLE 9 CONFUSION MATRIX OF RE-SAMPLED DATA – 1ST EXPERIMENT

FURIA				OneR			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
9	1	0	<i>a=no</i>	10	0	0	<i>a=no</i>
6	0	1	<i>b=alert</i>	2	5	0	<i>b=alert</i>
3	1	4	<i>c=alarm</i>	1	2	5	<i>c=alarm</i>

J48				Naïve Bayes			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
10	0	0	<i>a=no</i>	10	0	0	<i>a=no</i>
2	5	0	<i>b=alert</i>	3	3	1	<i>b=alert</i>
1	2	5	<i>c=alarm</i>	1	2	5	<i>c=alarm</i>

Neural Network			
Predicted class			
a	b	c	Real class
10	0	0	<i>a=no</i>
2	2	3	<i>b=alert</i>
0	1	7	<i>c=alarm</i>

TABLE 8 CLASSIFIER EVALUATION ON RE-SAMPLED DATA - 1ST EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	52	48	0.52	0.29	0.456	0.52	0.454	0.584
OneR	80	20	0.8	0.111	0.828	0.8	0.794	0.844
J48	80	20	0.8	0.111	0.828	0.8	0.794	0.826
NB	72	28	0.72	0.157	0.72	0.72	0.702	0.871
NN	76	24	0.76	0.138	0.784	0.76	0.76	0.80

From the results presented above it can be concluded that FURIA has the best prediction power on initial model of 1st experiment while on re-sampled data OneR and J48 have shown the highest predicting percentage.

2nd experiment

Attributes evaluation of data presented in 2nd experiment are shown in next table.

TABLE 10. ATTRIBUTES EVALUATION – 2ND EXPERIMENT

Attribute	Info Gain Attribute	Gain Ratio Attribute
	Eval	Eval
Previous temperature	0.0388	0.0167
Current temperature	1.0114	0.4356
time	0.043	0.043

Presented results show that the major impact to output variable (*fire*) has *current temperature* value.

Applying FURIA classifier to existing rules shown in Table 2, 50 rules are generalized into 7 presented in Table 11.

TABLE 11. THE FIRE DATA TEST OBTAINED USING FURIA IN 2ND EXPERIMENT

Previous temperature	Current temperature	time	Fire (class)
/	VL	/	no
/	L	/	no
/	M	L	no
/	H	L	alert
/	M	S	alert
/	VH	/	alarm
/	H	S	alarm

TABLE 12. CLASSIFIERS EVALUATION – 2ND EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	96	4	0.96	0.014	0.965	0.96	0.96	0.97
OneR	74	26	0.74	0.151	0.752	0.74	0.742	0.794
J48	96	4	0.96	0.014	0.965	0.96	0.96	0.96
NB	74	26	0.74	0.14	0.715	0.74	0.724	0.936
NN	100	0	1	0	1	1	1	1

J48 decision tree for presented fire data in 2nd experiment is shown in Fig. 8. The attribute with the maximum gain ratio, as it is showed in Table 10, is *current temperature* and it is selected as the splitting attribute.

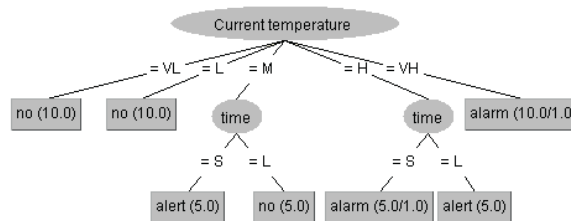


FIGURE 8 J48 DECISION TREE – 2ND EXPERIMENT

Classifiers evaluation is presented in Table 12.

TABLE 13. CONFUSION MATRICES – 2ND EXPERIMENT

FURIA				OneR			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
25	0	0	<i>a=no</i>	22	3	0	<i>a=no</i>
0	10	2	<i>b=alert</i>	5	6	1	<i>b=alert</i>
0	0	13	<i>c=alarm</i>	0	4	9	<i>c=alarm</i>

J48				Naïve Bayes			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
25	0	0	<i>a=no</i>	24	1	0	<i>a=no</i>
0	10	2	<i>b=alert</i>	4	4	4	<i>b=alert</i>
0	0	13	<i>c=alarm</i>	0	4	9	<i>c=alarm</i>

Neural Network			
Predicted class			
a	b	c	Real class
25	0	0	<i>a=no</i>
0	12	0	<i>b=alert</i>
0	0	13	<i>c=alarm</i>

TABLE 14 CLASSIFIERS EVALUATION ON RE-SAMPLED DATA – 2ND EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	92	8	0.92	0.031	0.92	0.92	0.92	0.941
OneR	92	8	0.92	0.033	0.923	0.92	0.919	0.944
J48	98	2	0.98	0.007	0.981	0.98	0.98	0.981
NB	90	10	0.9	0.041	0.907	0.9	0.898	0.964
NN	98	2	0.98	0.007	0.981	0.98	0.98	0.989

Results presented in Tables 12 and 13 show that Neural network classifier has the best prediction model. It generated a model with 100% correctly classified instances (CCI).

Results obtained by applying above mentioned algorithms on re-sampled data are given in Tables 14 and 15.

TABLE 15. CONFUSION MATRICES ON RE-SAMPLED DATA – 2ND EXPERIMENT

FURIA				OneR			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
22	0	0	<i>a=no</i>	22	0	0	<i>a=no</i>
0	13	2	<i>b=alert</i>	0	14	1	<i>b=alert</i>
0	2	11	<i>c=alarm</i>	0	3	10	<i>c=alarm</i>

J48				Naïve Bayes			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
22	0	0	<i>a=no</i>	22	0	0	<i>a=no</i>
0	14	1	<i>b=alert</i>	0	14	1	<i>b=alert</i>
0	0	13	<i>c=alarm</i>	0	4	9	<i>c=alarm</i>

Neural Network			
Predicted class			
a	b	c	Real class
22	0	0	<i>a=no</i>
0	14	1	<i>b=alert</i>
0	0	13	<i>c=alarm</i>

From obtained results it can be seen that Neural Network classifier and J48 decision tree generate the best prediction models on initial and on re-sample data but it is important to note that in this case, other algorithms also have high predictive power.

3rd experiment

Attributes evaluation of data presented in 3rd experiment are shown in Table 16.

TABLE 16. ATTRIBUTES EVALUATION – 3RD EXPERIMENT

Attribute	Info Gain Attribute Eval	Gain Ratio Attribute Eval
Temperature	0.21741	0.13717
Humidity	0.0042	0.00265
Light	0.08299	0.05236
CO	0.38509	0.24296

Presented results show that the major impact to output variable (*fire*) has *CO* and *temperature*.

Applying FURIA classifier to existing rules shown in Table 3, 81 rules are generalized into 13 presented in Table 17.

TABLE 17 THE FIRE DATA TEST OBTAINED USING FURIA IN 3RD EXPERIMENT

Temperature	Humidity	Light	CO	Fire (class)
L	L	/	L	no
M	/	L	L	no
L	M	/	L	no
L	H	L	/	no
L	/	/	M	alert
/	/	M	L	alert
H	/	L	L	alert
M	/	H	L	alert
M	/	L	M	alert
/	/	/	H	alarm
H	/	/	M	alarm
M	/	H	M	alarm
H	/	H	/	alarm

J48 decision tree for presented fire data is shown in Fig. 9. The attribute with the maximum gain ratio, as it is showed in Table 1, is *CO* and it is selected as the splitting attribute.

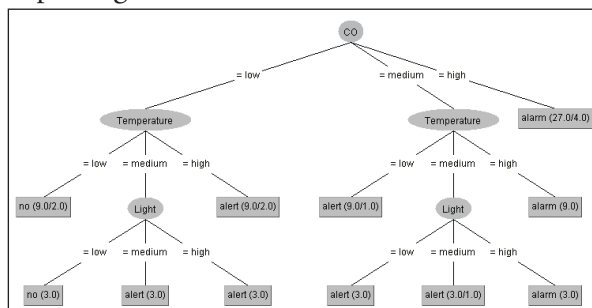


FIGURE 9 J48 DECISION TREE – 3RD EXPERIMENT

Classifiers evaluation is presented in Table 18.

In case of third experiment, Neural Network classifier has the best prediction model. It generated a model with 85.1% correctly classified instances (CCI), a precision of 85.9% (0.859) and the classification above the ROC curve area (0.951 > 0.5). Confusion matrices are presented in Table 19.

TABLE 19 CONFUSION MATRICES – 3RD EXPERIMENT

FURIA				OneR			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
8	3	1	<i>a=no</i>	0	10	2	<i>a=no</i>
2	16	13	<i>b=alert</i>	0	19	12	<i>b=alert</i>
0	3	35	<i>c=alarm</i>	0	15	23	<i>c=alarm</i>

J48				Naïve Bayes			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
9	2	1	<i>a=no</i>	3	8	1	<i>a=no</i>
0	24	5	<i>b=alert</i>	0	25	6	<i>b=alert</i>
0	4	34	<i>c=alarm</i>	0	6	32	<i>c=alarm</i>

TABLE 18 CLASSIFIERS EVALUATION – 3RD EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	72.8	27.1	0.728	0.203	0.732	0.728	0.716	0.847
OneR	51.8	48.1	0.519	0.344	0.457	0.519	0.482	0.587
J48	82.7	17.3	0.827	0.116	0.826	0.827	0.828	0.832
NB	74.1	25.9	0.741	0.184	0.778	0.741	0.723	0.872
NN	85.1	14.8	0.852	0.096	0.859	0.852	0.853	0.951

Neural Network			
Predicted class			
a	b	c	Real class
9	3	0	<i>a=no</i>
1	27	3	<i>b=alert</i>
0	5	33	<i>c=alarm</i>

Table 20 shows the predictive accuracy of the classifier on the re-sampled data. Obtained results show that all classifiers applied on re-sampled data have significantly better accuracy compared to results presented in Table 1. From Table 20 it can be seen that Neural Network classifier again has the best prediction model. On re-sampled data it generated a model with 93.8% correctly classified instances (CCI), a precision of 94% (0.94) and the classification above the ROC curve area (0.997 > 0.5).

Results shown in confusion matrices of re-sampled data in Table 21 are also better than ones presented in Table 19.

TABLE 21 CONFUSION MATRICES OF RE-SAMPLED DATA – 3RD EXPERIMENT

FURIA				OneR			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
14	2	2	<i>a=no</i>	14	1	3	<i>a=no</i>
1	13	5	<i>b=alert</i>	9	10	0	<i>b=alert</i>
0	2	42	<i>c=alarm</i>	2	8	34	<i>c=alarm</i>

J48				Naïve Bayes			
Predicted class				Predicted class			
a	b	c	Real class	a	b	c	Real class
14	1	3	<i>a=no</i>	12	6	0	<i>a=no</i>
2	15	2	<i>b=alert</i>	2	15	2	<i>b=alert</i>
0	2	42	<i>c=alarm</i>	0	5	39	<i>c=alarm</i>

Neural Network			
Predicted class			
a	b	c	Real class
17	1	0	<i>a=no</i>
1	17	1	<i>b=alert</i>
0	2	42	<i>c=alarm</i>

disparate and dynamic data, in real time or near real time. This reduces the transmission costs, and the data overload from a storage perspective.

The aim of this paper was to make a comparative analysis between different classification algorithms in a case of fire and to see which of applied techniques

TABLE 20 CLASSIFIER EVALUATION ON RE-SAMPLED DATA – 3RD EXPERIMENT

	CCI (%)	ICI (%)	TP	FP	P	R	FM	ROC
FURIA	85.1	14.8	0.852	0.121	0.852	0.852	0.849	0.947
OneR	71.6	28.4	0.716	0.117	0.747	0.716	0.724	0.8
J48	87.6	12.3	0.877	0.092	0.875	0.877	0.875	0.929
NB	81.5	18.5	0.815	0.078	0.843	0.815	0.822	0.95
NN	93.8	6.1	0.938	0.03	0.94	0.938	0.939	0.997

Obtained results show that Neural Network classifier generates the best prediction models on initial and on re-sample data.

CONCLUSION

Data mining in sensor networks is the process of extracting application-oriented models and patterns with acceptable accuracy from a continuous, rapid, and possibly non ended flow of data streams from sensor networks. The main purpose of sensors network for fire detection is to collect the monitoring original data, and provide basic information and decision support for monitoring centre. Also, data mining algorithm has to be sufficiently fast to process high-speed arriving data. In sensor networks, data are distributed by nature. The sensor scenario may often require in-network processing, wherein the data is processed to higher level representations before further processing. In other words, prediction in sensor networks can be performed in the way that each sensor learns a local predictive model for the global target classes, using only its local input data. On this way, individual nodes access and process local information and in order to achieve a collective decision, they must communicate to neighbor nodes, to send local and partial models and negotiate a common decision. In this case, whole data cannot be stored and must be processed immediately by their compressing and filtering for more effective mining and analysis in order to generate actionable insights from massive,

has the best prediction performances in order to reduce node activity and bandwidth.

FURIA was used as a base prediction model and it has shown the best prediction power in initial model of 1st experiment while on re-sampled data OneR and J48 obtained the highest predicting percentage. Neural Network classifier and J48 decision tree generated the best prediction models on initial and on re-sample data in 2nd experiment where all algorithms have shown high predictive power. Obtained results in 3rd experiment show that Neural Network classifier generates the best prediction models on initial and on re-sample data.

It can be seen that Neural Network classifier showed better predicting power on larger data set while in the case of small data set, simpler classifier like OneR or FURIA showed quite good results. Even applied data mining techniques are efficient, none of them can be considered as unique or general solution. On the contrary the selection of a correct data mining algorithm depends of an application and the compatibility of the observed data set. Thus, each situation should be considered as a special case and choice of adequate predictor or classifier should be performed very carefully based on empirical arguments.

Our future work will be based on measuring and combining real data from different sensors (temperature, humidity, light and CO) and selecting the best

prediction model for the given application classifying large data set at the sensor node level, discarding normal values and transmitting only anomaly values (*alert* and *alarm*) to the central server. This process by reducing the number of inputs, deleting redundancy, and improving the system speed and correct rate would decrease the potential network traffic and prolong network life span making early fire detection possible.

Authorship statement

Author(s) confirms that the above named article is an original work, did not previously published or is currently under consideration for any other publication.

Conflicts of interest

We declare that we have no conflicts of interest.

LITERATURE

- [1] Bahrepour, M., et al. (2009). Use of AI Techniques for Residential Fire Detection in Wireless Sensor Networks, AIAI-2009 Workshops Proceedings, pp. 311-321, Thessaloniki, Greece.
- [2] Bolourchi, P. and Uysal, S. (2013). Forest Fire Detection in Wireless Sensor Network Using Fuzzy Logic, Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), Madrid, Spain.
- [3] Chauhdary, S.H., et al. (2009). EOATR: Energy efficient object tracking by auto adjusting transmission range in wireless sensor network, *Journal of Applied Sciences*, 9(24), 4247-4252.
- [4] Gama, J. and Gaber, M.M. (2007). Learning from Data Streams, Processing Techniques in Sensor Networks, Springer-Verlag Berlin Heidelberg.
- [5] Han, J., et al. (2012). Data Mining: Concepts and Techniques, Morgan Kaufmann, USA.
- [6] Hühn, J. and Hüllermeier, E. (2009). FURIA: An Algorithm For Unordered Fuzzy Rule Induction, *Data Mining and Knowledge Discovery*, 19, 293-319.
- [7] Kapitanova, K., et al. (2011). Using fuzzy logic for robust event detection in wireless sensor networks, *Ad Hoc Netw.*
- [8] Kargupta, H., et al. (2010). A Vehicle Data Stream Mining System for Ubiquitous Environments, *Ubiquitous Knowledge Discovery Lecture Notes in Computer Science*, 6202:235-254.
- [9] Kumar, D. (2011). Monitoring forest cover changes using remote sensing and GIS: A global prospective, *Research Journal of Environmental Sciences*, 5 (2), 105-123.
- [10] Lee, L.T. and Chen, C.W. (2008). Synchronizing sensor networks with pulse coupled and cluster based approaches, *Information Technology Journal*, 7(5), 737-745.
- [11] Mahmood, A., et al. (2012). Mining Data Generated by Sensor Networks: A Survey, *Information Technology Journal*, 11(11): 1534-1543.
- [12] Manjunatha, P., et al. (2008). Multi-Sensor Data Fusion in Cluster based Wireless Sensor Networks Using Fuzzy Logic Method, IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, India.
- [13] McGarry, K. and MacIntyre, J. (2001). Data mining in a vibration analysis domain by extracting symbolic rules from RBF neural networks, Proceedings of the 14th International Congress on Condition Monitoring and Engineering Management, Manchester, U.K.
- [14] Perkowitz, M., et al. (2004). Mining models of human activities from the Web, Proceedings of the 13th International Conference on World Wide Web, New York, NY.
- [15] Rozyyev, A., et al. (2011). Indoor child tracking in wireless sensor network using fuzzy logic technique, *Research Journal of Information Technology*, 3(2), 81-92.
- [16] Sabri, N., et al. (2011). Wireless sensor actor network based on fuzzy inference system for greenhouse climate control, *Journal of Applied Sciences*, 11(17), 3104-3116.
- [17] Sathik, M.M., et al. (2010). Fire Detection Using Support Vector Machine in Wireless Sensor Network and Rescue Using Pervasive Devices, *International Journal of Advanced Networking and Applications*, 02(02), 636-639.
- [18] Tan, P.N. (2006). Knowledge Discovery from Sensor Data, *Sensors Magazine* (Cover story).
- [19] Tripathy, M.R., et al. (2010). Energy Efficient Fuzzy Logic Based Intelligent Wireless Sensor Network, Progress In Electromagnetics Research Symposium Proceedings, Cambridge, USA.
- [20] WEKA data mining tool, Available: www.cs.waikato.ac.nz/ml/WEKA.
- [21] Witten, I.H., et al. (2011). Data mining: practical machine learning tools and techniques, Morgan Kaufmann, Amsterdam
- [22] Yabukil, N., et al. (2011). Data Storage and Data Mining of Building Monitoring Data with Context, Proceedings of the 28th International Symposium on Automation and Robotics in Construction (ISARC2011), pp.377-378, Seoul, Korea.

Submitted: October 29, 2013.

Accepted: December 9, 2013.

ENUMERATION, RANKING AND GENERATION OF BINARY TREES BASED ON LEVEL-ORDER TRAVERSAL USING CATALAN CIPHER VECTORS

Adrijan Božinovski¹, Biljana Stojčevska², Veno Pačovski³

¹bozinovski@uacs.edu.mk, ²stojcevska@uacs.edu.mk, ³pachovski@uacs.edu.mk

Contribution to the state of the art

DOI: 10.7251/JIT1302078B

UDC: 579.253:577.2

Abstract: In this paper, a new representation of a binary tree is introduced, called the Catalan Cipher Vector, which is a vector of n elements with certain properties. It can be ranked using a special form of the Catalan Triangle designed for this purpose. It is shown that the vector coincides with the level-order traversal of the binary tree and how it can be used to generate a binary tree from it. Streamlined algorithms for directly obtaining the rank from a binary tree and vice versa, using the Catalan Cipher Vector during the processes, are given. The algorithms are analyzed for time and space complexity and shown to be linear for both.

The Catalan Cipher Vector enables a straightforward determination of the position and linking for every node of the binary tree, since it contains information for both every node's ancestor and the direction of linking from the ancestor to that node. Thus, it is especially well suited for binary tree generation. Using another structure, called a canonical state-space tableau, the relationship between the Catalan Cipher Vector and the level-order traversal of the binary tree is explained.

Keywords: Enumeration, Rank, Generation, Binary tree, Level-order traversal, Catalan Cipher Vector, Canonical State-Space Tableau

INTRODUCTION

Enumeration of binary trees means that every binary tree is linked to a unique linear representation, usually in a form of a sequence of integers or characters. Since there are C_n different binary trees of n nodes, with C_n being the n -th Catalan number, there should be C_n different representations to uniquely identify the trees. Every representation can be given a rank, usually a single integer, which establishes a relation of strict order between the binary trees. The conversion from a representation to a rank is usually done by using a Catalan Triangle, in a form that best suits the given representation.

The research in enumeration of binary trees has produced results including enumeration using bit strings [4, 8, 13] and integer sequences [5, 10, 11,

12]. Enumeration using integer sequences has been further subdivided into enumeration by Codewords [9, 14], weights [7] and distance [6]. More recently, enumeration using Catalan combinations [3] has been introduced.

The ranking of a binary tree is done by obtaining a unique value from the binary tree or its enumeration, which gives it a certain rank (i.e. number in a sequence) among other binary trees of a given size. The generation of a binary tree from its enumeration or rank represents the actual formation of the binary tree from its representation, whether it is the enumeration or the rank.

This paper introduces a new way of enumeration of a binary tree, called a Catalan Cipher Vector, and a way to transform that enumeration into previous

forms and vice versa. In particular, it will be shown how this enumeration relates to the level-order traversal of the binary tree. Streamlined algorithms will be presented, by which the rank of a binary tree is obtained from the generated binary tree and vice versa, during which the corresponding Catalan Cipher Vector elements will be obtained and directly utilized. The algorithms will be analyzed for time and space complexity.

INTRODUCING THE CATALAN CIPHER VECTOR

A Catalan Cipher Vector is the vector $v = [v_0 \ v_1 \ v_2 \ \dots \ v_{n-1}]$ which satisfies the following properties:

- 1) $v_0 = 0$;
- 2) $v_{i-1} + 1 \leq v_i \leq 2i$, for $i = 1, 2, 3, \dots, n - 1$ and $v_i \in \mathbb{N}$.

TABLE 1. LIST OF CODEWORDS, CATALAN COMBINATIONS AND CATALAN CIPHER VECTORS FOR $n = 4$

Rank	Codeword d	Catalan combination c	Catalan Cipher Vector v
0	[0 0 0 0]	[0 0 0 0]	[0 1 2 3]
1	[0 0 0 1]	[0 0 0 1]	[0 1 2 4]
2	[0 0 0 2]	[0 0 0 2]	[0 1 2 5]
3	[0 0 0 3]	[0 0 0 3]	[0 1 2 6]
4	[0 0 1 0]	[0 0 1 1]	[0 1 3 4]
5	[0 0 1 1]	[0 0 1 2]	[0 1 3 5]
6	[0 0 1 2]	[0 0 1 3]	[0 1 3 6]
7	[0 0 2 0]	[0 0 2 2]	[0 1 4 5]
8	[0 0 2 1]	[0 0 2 3]	[0 1 4 6]
9	[0 1 0 0]	[0 1 1 1]	[0 2 3 4]
10	[0 1 0 1]	[0 1 1 2]	[0 2 3 5]
11	[0 1 0 2]	[0 1 1 3]	[0 2 3 6]
12	[0 1 1 0]	[0 1 2 2]	[0 2 4 5]
13	[0 1 1 1]	[0 1 2 3]	[0 2 4 6]

$$\begin{aligned}
 d_0 &= 0; & c_0 &= 0; & v_0 &= 0; \\
 d_i &= c_i - c_{i-1} = v_i - v_{i-1} - 1; & c_i &= \sum_{j=0}^i d_j = v_i - i; & v_i &= \sum_{j=0}^i d_j + i = c_i + i; \\
 1 \leq i &\leq (n - 1), i \in \mathbb{N} & 1 \leq i &\leq (n - 1), i \in \mathbb{N} & 1 \leq i &\leq (n - 1), i \in \mathbb{N}
 \end{aligned}$$

Listing all distinct Catalan Cipher Vectors with lengths n shows that there are C_n such vectors. In Table 1, all Catalan Cipher Vectors are listed for $n = 4$, alongside the corresponding Codewords and Catalan combinations (the index of the first element in every representation is 0, and that element's value is also always 0). The relationships among the representations are also given at the bottom of the table. Unlike in the Codewords and the Catalan combinations, the elements in the Catalan Cipher Vectors are never equal to one another.

THE CANONICAL STATE-SPACE TABLEAU AND ITS CONNECTION TO THE CATALAN CIPHER VECTOR

The usefulness of the Catalan Cipher Vector can be demonstrated by using a special structure, a state-space tableau. In it, the binary tree is represented by using a tableau with dimensions $n \times 3$, where the first

(left-hand) column contains the symbols representing the values of the nodes of the tree, the second (middle) one contains the symbols representing the values of the nodes that the elements in the first column of the corresponding rows have as their left sub-nodes, and the third (right-hand) one contains the symbols representing the values of the nodes that the elements in the first column of the corresponding rows have as their right sub-nodes. In other words, the first column contains a node of the binary tree, the second column of that row contains that node's left sub-node (if the field is non-empty) and the third column of that row contains that node's right sub-node (if the field is non-empty). The root of the tree is the node of which the value is not found in the second or third column of the tableau (since the root does not have an ancestor node). Because a tree has at least one leaf, at least one of the rows in the tableau will have no values in the second and third column (since a leaf does not have any other nodes as sub-nodes). This is because a tree with n nodes has $n - 1$ edges: in a tableau of $3n$ fields, n are occupied in the first column and $n - 1$ in the second and third column, which leaves $n + 1$ empty fields; of those, at least two will be in the same row.

In the state-space tableau, it is irrelevant whether the rows are shuffled, since the structure of the binary tree is uniquely determined by the position relative to the columns, especially the second and third column, and not the rows. The row that has the elements in the second and third column as empty will be a leaf, and the element that is not found in the second and third column, but is present in the first one, will be the root. This is shown in Figure 1, where in all tableaux A is the root (it is present in the first, column, but not in the second and third column), while C and D are leaves (they have blank fields in the second and third column of their respective rows).

Since all state-space tableaux for a given binary tree are equivalent, it is necessary to choose one of them, to be worked with. The best choice is to select the tableau where the first row is the one that represents the root, and all other nodes are ordered in such a way that their vertical sequence in the first column follows the horizontal sequence in the second and

third column. Such a state-space tableau is called the *canonical state-space tableau*. For example, Figure 1b is the canonical state-space tableau for the binary tree in Figure 1a.

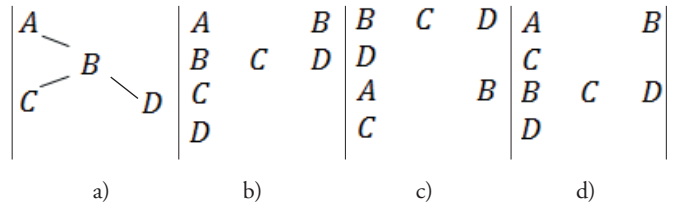
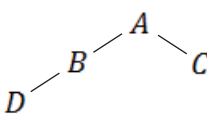
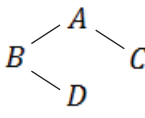
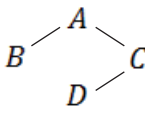
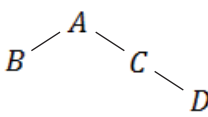
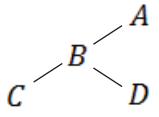
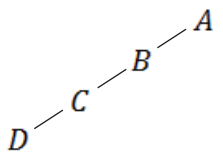
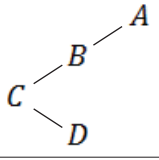
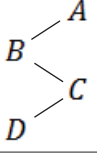
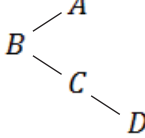
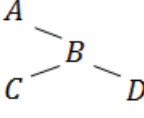


FIGURE 1. A) A BINARY TREE; B) ITS CANONICAL STATE-SPACE TABLEAU; C), D) OTHER EQUIVALENT STATE-SPACE TABLEAUX

Table 2 contains all binary trees for $n = 4$ and their corresponding canonical state-space tableaux and Catalan Cipher Vectors. For clarity, the elements in the canonical state-space tableaux are indexed, both with subscript and superscript indices. The subscripted indices, in the first column, represent the indices of the values stored in the nodes of the tree, obtained by following some traversal of the binary tree. The superscripted indices, in the second and third column, are sequential, starting with 1 at the top row in the second column and moving to the right and down, and enumerating only the elements in those two columns. If the field with a given index in the second or third column is non-empty, it represents the value stored in the node, which the node in the given row has as a left sub-node (if the field is in the second column, i.e. has an odd index) or right sub-node (if the field is in the third column, i.e. has an even index). An index of 0 denotes the root, and $v_0 = 0$ for every Catalan Cipher Vector, since the root does not have an ancestor. The elements of the Catalan Cipher Vector are also indexed with subscripted indices, to demonstrate the connection with the indices of the corresponding elements of the first column of the canonical state-space tableau.

TABLE 2. ALL BINARY TREES WITH THEIR CORRESPONDING CANONICAL STATE-SPACE TABLEAUX AND CATALAN CIPHER VECTORS FOR $n = 4$

Rank	Binary tree	Canonical state-space tableau	Catalan Cipher Vector ν
0		$\begin{matrix} A_0 & B^1 & C^2 \\ B_1 & D^3 & 4 \\ C_2 & 5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 2_2 \ 3_3]$
1		$\begin{matrix} A_0 & B^1 & C^2 \\ B_1 & 3 & D^4 \\ C_2 & 5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 2_2 \ 4_3]$
2		$\begin{matrix} A_0 & B^1 & C^2 \\ B_1 & 3 & 4 \\ C_2 & D^5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 2_2 \ 5_3]$
3		$\begin{matrix} A_0 & B^1 & C^2 \\ B_1 & 3 & 4 \\ C_2 & 5 & D^6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 2_2 \ 6_3]$
4		$\begin{matrix} A_0 & B^1 & 2 \\ B_1 & C^3 & D^4 \\ C_2 & 5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 3_2 \ 4_3]$
5		$\begin{matrix} A_0 & B^1 & 2 \\ B_1 & C^3 & 4 \\ C_2 & D^5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 3_2 \ 5_3]$
6		$\begin{matrix} A_0 & B^1 & 2 \\ B_1 & C^3 & 4 \\ C_2 & 5 & D^6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 3_2 \ 6_3]$
7		$\begin{matrix} A_0 & B^1 & 2 \\ B_1 & 3 & C^4 \\ C_2 & D^5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 4_2 \ 5_3]$
8		$\begin{matrix} A_0 & B^1 & 2 \\ B_1 & 3 & C^4 \\ C_2 & 5 & D^6 \\ D_3 & & \end{matrix}$	$[0_0 \ 1_1 \ 4_2 \ 6_3]$
9		$\begin{matrix} A_0 & 1 & B^2 \\ B_1 & C^3 & D^4 \\ C_2 & 5 & 6 \\ D_3 & & \end{matrix}$	$[0_0 \ 2_1 \ 3_2 \ 4_3]$

10		A_0 1 B^2 B_1 C^3 4 C_2 D^5 6 D_3	$[0_0 \ 2_1 \ 3_2 \ 5_3]$
11		A_0 1 B^2 B_1 C^3 4 C_2 5 D^6 D_3	$[0_0 \ 2_1 \ 3_2 \ 6_3]$
12		A_0 1 B^2 B_1 3 C^4 C_2 D^5 6 D_3	$[0_0 \ 2_1 \ 4_2 \ 5_3]$
13		A_0 1 B^2 B_1 3 C^4 C_2 5 D^6 D_3	$[0_0 \ 2_1 \ 4_2 \ 6_3]$

The benefit from the Catalan Cipher Vector is that it directly determines the topology of the binary tree, i.e. the connections between the nodes. The element with index 0 (i.e. the root) can have elements connected to it with values, in the Catalan Cipher Vector, of 1 and 2 only, which corresponds to its left and right sub-nodes, respectively, in the corresponding canonical state-space tableau. The element with index 1 can have elements connected to it with values, in the Catalan Cipher Vector, of 3 and 4 only etc. In other words, the element with index i can have connections to elements with values, in the Catalan Cipher Vector, of $2i + 1$ and $2i + 2$ only. This means that the index of the ancestor node for the node with index i (for $i \geq 1$) is directly obtainable from the corresponding value v_i of the Catalan Cipher Vector as $\lfloor \frac{v_i}{2} \rfloor$, if v_i is odd, or $\frac{v_i}{2} - 1$, if v_i is even. Furthermore, the direction of linking from the ancestor to the current node is also directly obtainable, since, following the canonical state-space tableau, the current node will be its ancestor's left sub-node if v_i is odd, or right sub-node if v_i is even.

As an example from Table 2, the tree with rank 12 has the Catalan Cipher Vector of $[0 \ 2 \ 4 \ 5]$. Viewing the node with index 1 (in this example, with information B), the value of the corresponding element from the Catalan Cipher Vector is $v_1 = 2$. Since v_1 is even, the node is its ancestor's right sub-node and its ancestor is the node with the index $\frac{v_2}{2} - 1 = 0$ (i.e. the

root, with information A). On the other hand, the value of the Catalan Cipher Vector for the node with index 3 (in this example, with information D) is odd ($v_3 = 5$), so the node is its ancestor's left sub-node and its ancestor is the node with the index $\lfloor \frac{v_3}{2} \rfloor = 2$ (i.e. the node with information C).

The Catalan Triangle

To establish a connection between the representation of the binary tree and its rank, several researchers [7, 11, 13] have utilized various forms of the Catalan Triangle. The form introduced in this paper is similar to the one used by [3], and is a transposed version of it. Figure 3 shows the Catalan Triangle that will be used in this paper, for $n = 4$. For clarity, the Catalan Triangle will be referred to as CT , and its elements will be given by their indices. For example, in Figure 2, $CT_{2,1} = 3$.

$i \downarrow j \rightarrow$	0	1	2	3
0	14			
1	9	5		
2	4	3	2	
3	1	1	1	1

FIGURE 2. A CATALAN TRIANGLE CT FOR $n = 4$

A closer examination of the Catalan Triangle in Figure 2 reveals that its elements can be obtained iteratively. Figure 3 presents the relationships among the elements of the Catalan Triangle in Figure 2.

$i \downarrow j \rightarrow$	0	1	2	3
0	$CT_{1,0} + CT_{1,1}$			
1	$CT_{2,0} + CT_{2,1}$	$CT_{2,1} + CT_{2,2}$		
2	$CT_{3,0} + CT_{3,1}$	$CT_{3,1} + CT_{3,2}$	$CT_{3,2} + CT_{3,3}$	
3	1	1	1	1

FIGURE 3. RELATIONSHIPS AMONG THE ELEMENTS OF THE CATALAN TRIANGLE FOR $n = 4$

In this Catalan Triangle, and any other for $n \geq 1$, all elements in the bottom row are 1, every element on the diagonal is the sum of the element below it and the element below and right to it, and every other element is a sum of the element below it and the element to the right of it. Therefore, the elements of the Catalan Triangle can be obtained as

$$CT_{i,j} = \begin{cases} 1 & i = n - 1 \\ CT_{i+1,j} + CT_{i+1,j+1} & i = j \\ CT_{i+1,j} + CT_{i,j+1} & \text{any other case} \end{cases}$$

An algorithm which initializes the Catalan Triangle using the aforementioned formula, for a given n , would be optimized for space and time, in a sense that the memory occupied would be only as much as needed, and that every element would be updated only once. Nevertheless, this means that there would be $n(n + 1)/2$ memory units utilized and as many time units for updating them, which would make such an algorithm $\mathcal{O}(n^2)$ for space and time complexity. However, once the Catalan Triangle would be initialized, all other algorithms would utilize the information stored in it and their performances would be enhanced.

Algorithms for Converting between the Rank and the Binary Tree

In this paper, the algorithms will be presented in pseudocode that resembles the C++ programming language, where the keywords will be displayed in italic type. The keyword *ref* means that the value is passed by reference, i.e. that it will be modified within the algorithm and that its modified version will be available after the algorithm ends. The keyword *define* is used to define auxiliary variables or references within the algorithm. The keyword *new* means that memory will be assigned to the reference it affects. The keyword *array* means that the memory assigned to the reference will be an array with a specified size. The keyword *node* means that the memory

assigned to the reference will be of a type of a binary tree node. The keyword *isEmpty* is an algorithm that returns true if a queue is empty and false otherwise. The keywords *enqueue* and *dequeue* are algorithms for enqueueing into or dequeueing from a queue, respectively. The keywords *leftSubNode* and *rightSubNode* refer to a left sub-node and a right sub-node of a given node, respectively. Other keywords, as well as special characters, retain their corresponding meanings and functions from the C++ language.

An efficient algorithm for obtaining a Catalan combination from a given rank is given in [3]. It can be slightly modified to produce a corresponding Catalan Cipher Vector as a result, because of the interchangeability between a Catalan combination and a Catalan Cipher Vector (Table 1). Since each element of the Catalan Cipher Vector determines the predecessor of the corresponding node in the binary tree, as well as whether it is its predecessor's left or right sub-node, the binary tree can be generated immediately after obtaining each of its elements. Since the root does not have a predecessor, it can be generated directly, without linking it to any other node.

Algorithm rank2tree (Figure 4a) shows how to obtain the binary tree for a given rank. First the Catalan Cipher Vector element is obtained and then a new node of the binary tree is generated, for which the index of its predecessor and the direction of linking from it is calculated, based on the value of the element of the vector. If the current node is the root, no linking takes place; else, the current node is linked to the predecessor node. All nodes are generated in an array, and only the first node of the array is returned, which is the root of the tree.

Algorithm val (Figure 4b) is an auxiliary algorithm that generates values of the information fields for the nodes of the trees based on their indices, which in this case is set to return the index itself as the information field of the node. Arbitrary logic can be used if it needs to return different information fields based on the indices (for example, the trees in Figures 1 and 2 and Table 1 have the letters of the alphabet stored in the information fields of the nodes).

```

rank2tree(rank, CT, n){
  define i, base, v, tn;
  i = 0; base = 0;
  v = new array(n);
  tn = new array(n);
  tn[0] = new node(val(0));
  while(i < n){
    while((base < n) && (CT[i][base] <= rank)){
      rank = rank - CT[i][base];
      base++;
    }
    v[i] = base+i;
    tn[i] = new node(val(i));
    if(i != 0)
      if(v[i]%2 != 0)
        tn[v[i]/2].leftSubNode = tn[i];
      else
        tn[v[i]/2 - 1].rightSubNode = tn[i];
    i++;
  }
  return tn[0];
}

```

a)

```

val(ind){
  return ind;
}

```

FIGURE 4. A) AN ALGORITHM FOR OBTAINING THE BINARY TREE OF A GIVEN RANK; B) THE AUXILIARY ALGORITHM FOR OBTAINING THE VALUE OF A NODE OF THE BINARY TREE, BASED ON ITS INDEX

The time complexity analysis of rank2tree is concerned with the overhead from calculating the value of an individual Catalan Cipher Vector element, based on the rank, by traversing the Catalan Triangle. In the worst case [3] it takes $2n - 1$ time units and in the best case it takes $n - 1$ time units to obtain an individual element of the vector from the given rank. Since each of the n nodes of the tree will be generated once (and since the algorithm val is $\Theta(1)$), this means that the overall time complexity of rank2tree is $\Theta(n)$. For the space complexity, it can be seen that the only memory elements with variable sizes are the two arrays that represent the intermediate Catalan Cipher Vector and the nodes of the binary tree respectively. Therefore, the space complexity of this algorithm is also $\Theta(n)$.

To obtain the rank for a given binary tree, first the Catalan Cipher Vector would need to be obtained from the tree and then the total rank would be calculated by the contribution from each element of the vector. The Catalan Cipher Vector is linked with the canonical state-space tableau in which the left and right sub-nodes for every node are represented. To obtain each element of the Catalan Cipher Vector from each node of the tree, the tree would need to be traversed in a way so that, for a given root, first the left sub-node would be traversed and then the right sub-node, but without going into recur-

sion, i.e. the traversal would be by depth. By definition [2], this is level-order traversal and it is therefore used as the traversal of choice for obtaining each element of the Catalan Cipher Vector for each node.

Algorithm tree2rank (Figure 5a) shows how each element of the Catalan Cipher Vector is obtained following the level-order traversal of the tree. When it is obtained, it is used to update the overall rank by using algorithm update (Figure 6b). Since tree2rank incorporates the level-order traversal of a tree, which is a $\Theta(n)$ algorithm, its time complexity analysis requires an analysis of algorithm update. update contains a loop which runs only if displacement occurs, of the column of the Catalan Triangle under consideration. This happens when there is a change in the value of an element of the Catalan combination ($c_i = v_i - i$) relative to the previous element ($c_{i-1} = v_{i-1} - (i - 1)$) and the loop will not run if those two elements are identical. The numbers of displacements from the previous to the next element in the Catalan combination (or Catalan Cipher Vector) are actually the elements of the corresponding Codeword, and the sum of all elements in the Codeword for a given n is at most $n - 1$ (examples for $n = 4$ are given in Table 1). Thus, when no displacement occurs, there are n movements through the Catalan Triangle, and this is the best case, i.e. the algorithm is $\Omega(n)$. The worst-case scenario occurs when there are $n - 1$ displacements in the Catalan Triangle, which means there are $n + (n - 1) = 2n - 1$ movements, i.e. the algorithm is $O(n)$. This means that the time complexity of update is $\Theta(n)$, so the overall time complexity of tree2rank is also $\Theta(n)$.

For the space complexity analysis, it can be seen that tree2rank requires an array v of n integers, which represents the resulting Catalan Cipher Vector, and a queue q of nodes, which will occupy various amounts of memory, depending on the tree being traversed. Assuming that each node occupies k units of memory, and the integers each take a unit of memory, the space complexity can be analyzed. In the best case, one of a degenerate tree, there will be only one node in the queue during the level-order traversal, which means that the total memory usage will be $n + k$ and thus the space complexity will be $\Omega(n)$. In the worst case, that of a complete binary tree, for every node dequeued two more nodes will be enqueued, so there will be at

most $\frac{n+1}{2}$ nodes in the queue and thus the full memory usage will be $n + k \frac{n+1}{2} = n \left(1 + \frac{k}{2}\right) + \frac{k}{2}$, i.e. the space complexity will be $O(n)$. Therefore, the space complexity will also be $O(n)$.

value of its index, or every $v_i = i$, for $0 \leq i \leq n - 1$. In the canonical state-space tableau, this corresponds with a tableau where the second and third column would be filled up as follows: first the second column of the first

```
tree2rank(t, CT, n){
    define q, v, i, row, rank, p;
    v = new array(n);
    i = 1;
    row = 0;
    rank = 0;
    q.enqueue(t);
    v[0] = 0;
    while(!q.isEmpty()){
        p = q.dequeue();
        if(p.leftSubNode != NULL){
            q.enqueue(p.leftSubNode);
            v[i] = 2*row + 1;
            update(rank, CT, v, i);
            i++;
        }
        if(p.rightSubNode != NULL) {
            q.enqueue(p.rightSubNode);
            v[i] = 2*row + 2;
            update(rank, CT, v, i);
            i++;
        }
        row++;
    }
    return rank;
}
```

a)

```
update(ref rank, CT, v, i){
    define prev, next, j;
    prev = v[i-1]-(i-1);
    next = v[i]-i;
    for(j = prev; j < next; j++){
        rank = rank + CT[i][j];
    }
}
```

b)

FIGURE 5. A) AN ALGORITHM FOR CALCULATING THE RANK FROM A GIVEN BINARY TREE. B) AN AUXILIARY ALGORITHM FOR UPDATING THE RANK GIVEN THE CATALAN CIPHER VECTOR AND THE INDEX OF THE ELEMENT WHICH CONTRIBUTES TO THE OVERALL RANK

Special Forms of the Binary Trees Obtained from Certain Ranks

The goal of the enumeration of binary trees is to establish a 1-to-1 relation between a binary tree and its representation. In this paper, every binary tree has a unique Catalan Cipher Vector and a unique rank related with it. The advantage of the ranking system presented in this paper is that certain ranks always produce certain forms of the binary trees. It can be viewed in Table 2, for example.

The rank of 0 is equivalent to the initial Catalan Cipher Vector, where each element of the vector has the

row, then the third column of the first row, then the second column of the second row, then the third column of the second row and so on. In other words, every node will obtain first a left then a right sub-node, before the same thing happens for the next node in the traversal. Since the traversal is level-order, the levels of the tree will be filled from left to right and the nodes will have both left and right sub-nodes, or only a left sub-node (if there is an even number of nodes; there will be only one such node), or no sub-nodes (such will be the leaves). This means that all levels will be completely filled, except the last level, where all nodes will be positioned as far left as possible. This is the definition of a complete binary tree [1]. Therefore, it can be said that algorithm rank2tree on Figure 5 will create the complete binary tree of n nodes, if the given rank is 0. Equivalently, a tree of rank 0 is a complete binary tree for any number of nodes n , when using

algorithm rank2tree. For $n = 2^k - 1, k \geq 1, k \in \mathbb{N}$, the tree with rank 0 is a full binary tree.

The rank of $C_n - 1$ yields a Catalan Cipher Vector where $v_i = 2i$, for $0 \leq i \leq n - 1$. In the canonical state-space tableau, this corresponds with a tableau where only the third column is filled for every row except the last. In the generated tree, this means that every node (except the last) will have only a right sub-node, which is a feature of a degenerate tree. Therefore, it can be said that algorithm rank2tree on Figure 5 will create a degenerate tree of n nodes, if the given rank is $C_n - 1$. Equivalently, a tree of rank $C_n - 1$ is a degenerate tree for any number of nodes n , when using algorithm rank2tree.

CONCLUSION

A new way of enumerating binary trees, called the Catalan Cipher Vector, is introduced. Another representation, the canonical state-space tableau, helps to show how the Catalan Cipher Vector determines the

entire topology of the binary tree. Algorithms are given which show how to obtain the rank from a given binary tree and vice versa, using the Catalan Cipher Vector within the algorithms themselves. It is shown how, following those algorithms, certain ranks always produce certain forms of the binary trees. Since the algorithms for conversion from a binary tree to its respective rank and vice versa are linear in both time and space complexity, while utilizing the Catalan Cipher Vector as an intermediate result, it is an efficient representation. To the best of the knowledge of the authors, the Catalan Cipher Vector is the first enumeration that utilizes level-order traversal to generate the binary tree from itself, and it is their belief that it is therefore also more intuitive and elegant.

Authorship statement

Author(s) confirms that the above named article is an original work, did not previously published or is currently under consideration for any other publication.

Conflicts of interest

We declare that we have no conflicts of interest.

REFERENCES

- [1] Black, P. E. "complete binary tree", in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 26 May 2011. Available from: <http://www.nist.gov/dads/HTML/completeBinaryTree.html>. Accessed on July 4, 2013
- [2] Black, P. E. "level-order traversal", in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 26 May 2011. Available from: <http://www.nist.gov/dads/HTML/levelOrderTraversal.html>. Accessed on July 30, 2013
- [3] Črepinšek, M. and Mernik, L. (2009). An Efficient Representation for Solving Catalan Number Related Problems. *International Journal of Pure and Applied Mathematics*, Vol. 56, No. 4, 589-604.
- [4] M. C. Er. (1985). Enumerating Ordered Trees Lexicographically. *The Computer Journal*, Vol. 28, No. 5, 538-542.
- [5] Knott, G. D. (1977). A Numbering system for binary trees. *Communications of the ACM*, Vol. 20, No.2, 113-115.
- [6] Mäkinen, E. (1987). Left Distance Binary Tree Representations. *BIT Numerical Mathematics*, Vol. 27, No. 2, 163-169.
- [7] Pallo, J. M. (1986). Enumerating, Ranking and Unranking Binary Trees. *The Computer Journal*, Vol. 29, No. 2, 171-175.
- [8] Proskurowski, A. (1980). On the generation for binary trees. *Journal of the ACM* Vol. 27, 1-2.
- [9] Roelants van Baronaigen, D. (1991). A loopless algorithm for generating binary tree sequences. *Information Processing Letters* 39, 189-194.
- [10] Rotem, D. and Varol, Y. L. (1978). Generation of binary trees from ballot sequences. *Journal of the ACM*, Vol. 25, No.3, 396-404.
- [11] Ruskey, F. and Hu, T. C. (1977). Generating Binary Trees Lexicographically. *SIAM Journal on Computing*, Vol. 6, No. 4, 745-758.
- [12] Xiang, L., Tang, C. and Ushijima, K. (1997). Grammar-Oriented Enumeration of Binary Trees. *The Computer Journal*, Vol. 40, No. 5, 278-291.
- [13] Zaks, S. (1980). Lexicographic Generation of Binary Trees. *Theoretical Computer Science*, Vol. 10, 63-82.
- [14] Zerling, D. (1985). Generating binary trees using rotations. *Journal of the ACM*, Vol. 27, 694-701.

Submitted: July 31, 2013.

Accepted: December 23, 2013.

USING DECISION TREE CLASSIFIER FOR ANALYZING STUDENTS' ACTIVITIES

Snježana Milinković¹, Mirjana Maksimović²

¹snjeza@etf.unssa.rs.ba, ²mirjana@etf.unssa.rs.ba

Case study

DOI: 10.7251/JIT1302087M

UDC: 37.018.43:004.738.5

Abstract: *In this paper students' activities data analysis in the course Introduction to programming at Faculty of Electrical Engineering in East Sarajevo is performed. Using the data that are stored in the Moodle database combined with manually collected data, the model was developed to predict students' performance in successfully passing the final exam. The goal was to identify variables that could help teachers in predicting students' performance and making specific recommendations for improving individual activities that could directly influence final exam successful passing. The model was created using decision tree classifier and experiments were performed using the WEKA data mining tool. The effect of input attributes on the model performances was analyzed and applying appropriate techniques a higher accuracy of the generated model was achieved.*

Key Words: *decision tree, moodle, students' performances, e-learning*

INTRODUCTION

The process of knowledge acquiring and transmitting was dramatically changed by the progress in the use of information - communication technologies. Electronic learning (e-learning) has become an area where significant research efforts have been invested with aim to improve existing and find new and attractive method of knowledge dissemination. The basic tendency is to increase the motivation of e-learning courses' users and achieving the best possible outcomes. Learning Management Systems - LMS are software applications used for creation, organization and administration of e-learning courses. These softwares are specially designed for educational purposes and their applications provide user-friendly access to learning contents, easy creation and presentation of learning material, interactive communication among users, testing and polling of users, assessment activities, and so on. One of the LMSs that is widely used in academic communities around the world is Moodle (Modular Object-Oriented Dynamic Learning Environment) [5].

Moodle allows easily creation of electronic courses and adaption of traditional course to formats suitable for e-learning. In addition, it allows tracking all the activities of its users. The information about each user's activities is kept in the Moodle database of Moodle system and it is available to the system administrators at any time. This functionality option of Moodle application is very important because of vast amounts of potentially useful data that are accumulated in this way.

Applying suitable transformation and discretization techniques on data obtained from a Moodle, which can be generated from various reports on activities, it is possible to obtain a form that is suitable for the application of data mining algorithms [9]. Data mining is usually defined as the process of discovering useful patterns or knowledge from different data sources [4]. The main goal of data mining techniques is to find and describe the structural patterns in the data in order to attempt to explain connections between data and create predictive models

based on them [13]. Data mining is a multidisciplinary field which includes machine learning, statistics, databases, artificial intelligence, information theory and visualization [4]. One of the most common tasks used in data mining applications is the classification. Classification is type of machine learning analogue to human learning from past experiences to gain new knowledge in order to improve our ability to perform real-world tasks [4]. Computers using machine learning learns from data which are collected in the past and represent past experiences. In most cases classification is used for learning a target function that can be used to predict the values of a discrete class attribute, e. g. classification is one type of predictions methods. The goal of prediction is to infer a target attribute, predicted variable, from some combination of other aspects of the data or another attribute. Classification here means the problem of correctly predicting the probability that an example has a predefined class from a set of attributes describing the example. In classification learning, the learning scheme is presented with a set of classified examples from which it is expected to learn a way of classifying unseen examples [13]. The process of data mining consists of three basic steps:

- Pre-processing – the raw data must be cleaned in order to become suitable for mining. Data cleaning includes removing noises and abnormalities, handling too large data, identifying and removing irrelevant attributes, and so on. Data cleaning is procedure that usually consumes a lot of time and it is very labor-intensive but it is absolutely necessary step for successful data mining.
- Application of data mining algorithms – the process of applying data mining algorithm that will produce patterns or knowledge.
- Post-processing – Among all discovered patterns or knowledge, it is necessary to discover ones that are useful for the application. For making the right decision there are many evaluation and visualization techniques that can be used.

Data mining can be applied to research and analyze the data that come from educational environments. This new developing field, known as Educational Data Mining (EDM), began to develop

intensively in recent years. It is engaged in the development of methods for exploring the unique types of data that come from the educational context [10]. The main objective is to discover the implicit and useful patterns or knowledge about how students learn and the factors that affect their learning. Gained knowledge can be used to provide feedbacks to the teachers in order to improve the teaching process through more quality and easier management of students in the learning process achieving the best possible outcomes.

In recent years, a lot of research in the field of educational data mining was performed. An overview of the current state and the progress made in the development and implementation of educational data mining is given in [10]. Prediction of the achieved success and the final grade in the exam can be performed applying data mining algorithms. In [1], the ranking of factors that influence the prediction of academic performance in order to identify students who will need to study harder to pass the exam was performed by the application of data mining methods. An experiment with pattern classification for student performance prediction is performed in [2]. The obtained results illustrate that recognition for a certain class on a large data set can be obtained by a classifier built from a small size data set. The scope of [7] was to identify the factors influencing the performance of students in final examinations and find out a suitable data mining algorithm to predict the grade of students. The obtained results reveals that type of school does not have influence on students' performance while parents' occupation plays a major role in predicting grades. The focus of the research can be put on usage of data mining methods for analyzing the quality and methods that e-learning courses content is presented to the students [6]. The impact of the certain e-learning tools on the achievement of students' objectives is discussed in [3]. In [8] a survey about the application of data mining to web-based electronic courses and learning content management systems was performed. As a result, a general model that represents the whole process of application of data mining techniques in educational systems was created (Figure 1). A specific Moodle mining tool oriented for the use of not only experts in data mining is described in [11]. Also, the performances of differ-

ent data mining techniques for classifying students are compared. Performed experiments show that in general there is not one single algorithm that obtains the best classification accuracy in all cases while some pre-processing task like filtering, discretization or re-balancing can be very important to obtain better or worse results.

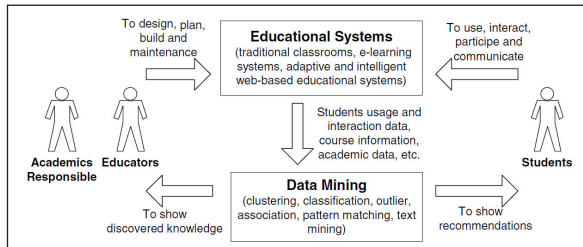


FIGURE 1 APPLICATION OF DATA MINING IN EDUCATIONAL SYSTEMS

This paper analyzes the impact of specific pre-exam activities on actual student performance in the course Introduction to programming that is performed in Faculty of Electrical Engineering in East Sarajevo. A model for predicting students' performance in the final exam was developed by analyzing course activities. Most of the data about these activities were stored in the Moodle database while some of the data were manually collected (students' attendance on lectures). From the Moodle database, a randomly selected data for one generation of students were chosen. The model was created using a decision tree classifier. Presented experiments were performed using WEKA data mining tool [12]. The influence of input attributes on the performance of the model was analyzed and higher accuracy of the generated model was achieved by application of appropriate techniques.

The rest of this paper is organized as follows. The course organization, data collection and preprocessing are described in second section. Third section presents J48 Decision tree algorithm while in fourth section simulation results of four proposed experiments are shown. Finally, fifth section provides conclusion remarks and outlines directions for future work.

COURSE ORGANIZATION, DATA COLLECTION AND PREPROCESSING

For the purposes of this study, data about students who have attended the Introduction to Pro-

gramming course, which is performed during the summer semester of the first year of study in Electrical Engineering in East Sarajevo, were collected and analyzed. Randomly sampling, the data of the students from all three study programs that are running at the faculty were collected. Electronic course was implemented as a complement to the traditional way of teaching what means that concept of blended learning is applied. The main objective of the electronic course creation is to improve the efficiency of traditional ways of teaching. The course was created using the Moodle platform and has been used to provide various learning resources and facilitate communication among its participants. Traditional course content, and thus supporting electronic course, was organized through three parts: lecture, problem solving exercises and laboratory exercises. Through the pre-exam activities, students are required to attend and successfully complete 3 cycles of laboratory exercises. Other activities in the course (homework assignments, successfully done tasks on preparatory cycle of laboratory exercises, lessons access, forums, and access to other resources of course) are not part of the mandatory pre-exam activities but some of them are scored. The number of points gained through these non-mandatory activities represents bonus in addition to the maximum required number of points which student can earn. A percentage values of the successful completion of certain activities are stored in the Moodle database for each student, as it shown in Figure 2.

Мogućа оцена	Оцена	Опсег	Процент	Покривна информација
Увод у програмирање				
Лаб. вежба 8 - 9. април 2013.	4,00	0-7	57,14 %	
Лаб. вежба 8 - 9. 4. 2013. - прилози датума	5,00	0-7	71,43 %	
Лаб. вежба - 12. 04. 2013.	10,00	0-10	100,00 %	
Upload zadaca	-	0-10	-	
вежбене изолаци	9,00	0-10	90,00 %	
вежбене матурице	4,00	0-6	66,67 %	
Лаб. вежба 07. 06. 2013.	10,00	0-10	100,00 %	
Лаб. вежба 10. 06. 2013.	10,00	0-10	100,00 %	
Укупно за курс	86,67	0-100	86,67 %	

FIGURE 2 A PERCENTAGE VALUES OF THE SUCCESSFUL COMPLETION OF CERTAIN ACTIVITIES FOR ONE STUDENT

This information is extracted from the Moodle database for randomly chosen students. Manually collected data about points that students earned by attendance on lectures during semester are added to this information and together they present input data for data mining process. To be able to apply

data mining techniques, it is necessary to preprocess input data. In the initial stage of preprocessing, data of students who have not obtained the minimum points required for the successful defense of mandatory laboratory exercises are discarded. Next step is identifying and discarding the attributes that have no predictive value (the index number, name, and so on). After that, all percentage values extracted from Moodle database are recalculated in the number of points for particular activities. By manually discretization process [11] a numerical values which represent the final grade of class attribute 'results' were transformed into nominal values in accordance with the specific needs of the individual experiments performing. After that, using the filter method Info-gain the values of input attributes in relation to the class attribute have been evaluated. In this manner, in data set used in the study, attributes that have no impact on values of the class attribute are identified and discarded. All discarded attributes in this preprocessing step belonged to the set of attributes that describe the non-mandatory course activities (graded and ungraded).

Data preprocessing is a procedure that usually consumes a bulk of time and requires a lot of work, but it is an absolutely necessary step for the successful application of data mining techniques and algorithms.

J48 DECISION TREE ALGORITHM

The decision tree is a very popular method for classification and decision making. It is a decision making technique based on the relationship between strategy and conditions, and it is used to solve many problems. It predicts outcomes using a series of questions and rules for data classification. The decision tree branching occurs as a result of meeting the requirements of classification issues. Each question will divide data into subsets that are more homogeneous than the senior set. If the question has two answers, then the response to the question arise two subsets (binary tree). Subsets arise according to number of questions answers. Therefore the classification of certain data are carried out. Predicting the behavior of a particular client can be made on the basis of its belonging to a particular event (which is classified

based on a number of issues and conditions), for which we know how it acts. During the construction of decision trees is important to know the right questions. The main advantage of decision tree classifier is its classification speed. The models which are based on the decision tree algorithms differ in certain data characteristics which are required and in which basis issues are created [13]. In this paper, J48 decision tree, which is an implementation of C4.5 algorithm in WEKA data mining tool [12], is used.

SIMULATION RESULTS

In order to obtain as much useful information of the individual attributes impact on students' performance in the course Introduction to Programming with aim of obtaining a large percentage of correctly classified instances, the work presented in this paper is carried out through several experiments:

- 1st experiment:
- Used attributes are: laboratories (total), student attendance on lectures and results (passed and failed).
- 2nd experiment:
- Used attributes are: laboratory exercises of first, second and third cycle (L1, L2 and L3, respectively), student attendance on lectures and results (passed and failed).
- 3rd experiment:
- Used attributes are: laboratories (total), student attendance on lectures and results (passed in June-July period, passed in other periods and failed).
- 4th experiment:
- Used attributes are: laboratory exercises of first, second and third cycle (L1, L2 and L3, respectively), student attendance on lectures and results (passed in June-July period, passed in other periods and failed).

Attribute 'results' in all 4 experiments is referred to as a class variable.

1st experiment

Attributes evaluation can be performed using Info-Gain Attribute Evaluation and Gain-Ratio Attribute Evaluation. Info-Gain evaluates attributes by measuring their information gain with respect to

the class. This method can treat missing as a separate value or distribute the counts among other values in proportion to their frequency. Gain-Ratio Attribute Evaluation evaluates attributes by measuring their gain ratio with respect to the class. Attributes with estimates of less than 0.01 should be excluded from the analyzed data set. For attributes proposed in 1st experiment, their evaluation is given in Table 1.

TABLE 1. ATTRIBUTES EVALUATION – 1ST EXPERIMENT

Attribute	InfoGain AttributeEval	GainRatio AttributeEval
Lab.	0.203	0.208
Attendance	0.125	0.126

Table 1 show that laboratory exccerics have the major impact to final results. The attribute with the maximum gain ratio is selected as the splitting attribute what can be seen from Figure 3 a.

In the first experiment, after applying the J48 classifier an accuracy of 71.8 % is achieved and created tree is shown in Figure 3 a. The numbers given in parentheses are the number of instances assigned to that node number followed incorrectly classified instances. The minimum number of instances per node (minNumObj) was kept at 2 and during the experiment 10-fold cross-validation is applied, which is a standard method for predicting the error rate learning techniques of a given fixed sample of data. The data were divided into 10 subsets where classes are represented in approximately the same proportion as in the full data set. Each part is done in order and learning scheme is trained on the remaining nine-tenths, and the error rate is calculated on the set of the test sample. Thus, the learning is performed a total of 10 times on different training sets (each set has much in common with the other). Finally, there is an average value of 10 estimated errors to obtain an estimate of the total error [13]. If the minimum number of instances per node (minNumObj) is increased to 3 a simpler tree shown in Figure 3 b is obtained, but the accuracy of correctly classified instances is less than the previous case -70.4%.

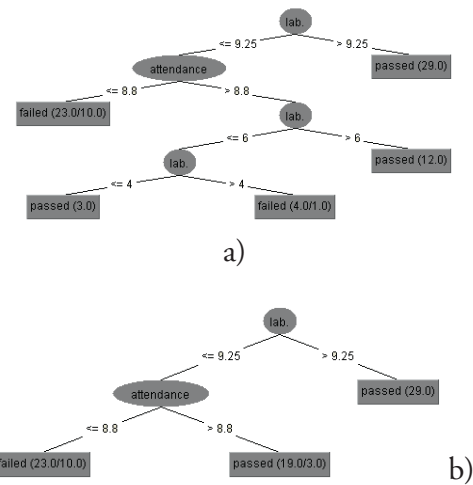


FIGURE 3 DECISION TREE (1ST EXPERIMENT): A) THE INITIAL MODEL, B) THE MODEL WITH INCREASED MINIMUM NUMBER OF INSTANCES PER NODE

2nd experiment

Results of attributes evaluation in second experiment are shown in Table 2.

TABLE 2. ATTRIBUTES EVALUATION – 2ND EXPERIMENT

Attribute	InfoGain AttributeEval	GainRatio AttributeEval
L1	0	0
L2	0.226	0.258
L3	0.193	0.219
Attendance	0.126	0.125

Table 2 shows that the attribute with the maximum gain ratio is L2 and it is selected as the splitting attribute while first laboratory exercise L1 evaluation is less than 0.01 so it should be excluded from the dataset.

Observing the effects of individual laboratory exccerics (L1, L2 and L3) and the student attendance on lectures on achieved results at the exam in the second experiment, using the J48 decision tree and the 10-fold cross-validation, obtained accuracy is 74.6 %. Created tree is shown in Figure 4 a.

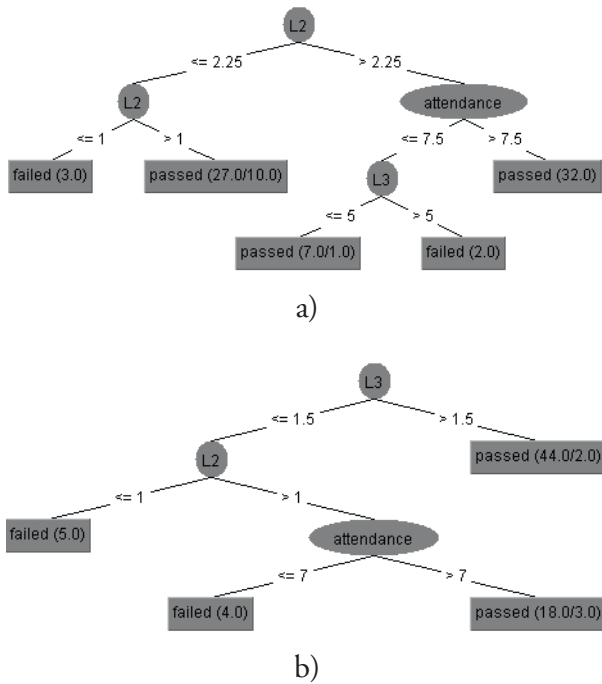


FIGURE 4 DECISION TREE (2ND EXPERIMENT): A) THE INITIAL MODEL, B) THE MODEL ACHIEVED OVER A BALANCED DATA

In multiclass prediction, the result on a test set is often displayed as a two-dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements. The results are shown in Table 3.

TABLE 3. CONFUSION MATRIX

Predicted class		Real class
a	b	
50	5	<i>a=pass</i>
13	3	<i>b=failed</i>

Originally generated model have shown unbalanced distribution of examples per class variables, what indicated that the data were not well prepared. In the case of unbalanced data sets, examples of small classes are more difficult to train. The problem with unbalanced data arises because learning algorithms tend to overlook less frequent classes (minority classes), paying attention just to the most frequent ones (majority classes). As a result, the classifier obtained is not able to correctly classify data instances corresponding

to poorly represented classes. One of the most frequent methods used to learn from unbalanced data consists of re-sampling the data. To solve this problem, in this paper resampling was performed using Resample Weka filter for supervised learning with or without instance replacement. Created decision tree on re-sampled data is shown in Figure 4 b. Achieved accuracy in this case is 87.3%. It can be concluded that accuracy is significantly increased on re-sampled data and the decision tree classifier created more precise model.

3rd experiment

In the third experiment, the emphasis is on the impact of attributes laboratory exercises (total) and student attendance on lectures on passing the exam in the first, June-July, final exam term.

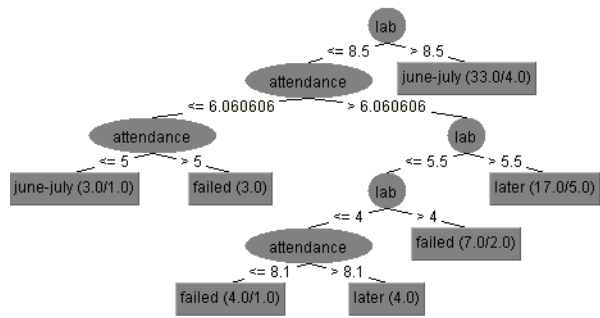
For those proposed attributes their evaluation is given in Table 4.

TABLE 4. ATTRIBUTES EVALUATION – 3RD EXPERIMENT

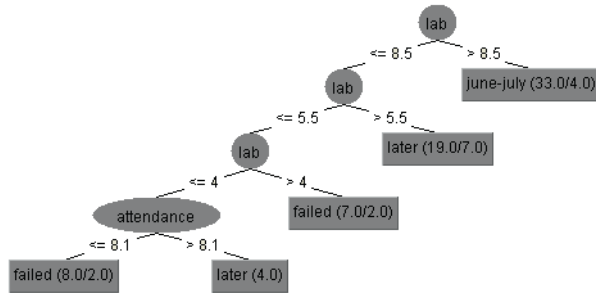
Atribut	<i>InfoGain</i>	<i>GainRatio</i>
	<i>AttributeEval</i>	<i>AttributeEval</i>
Lab.	0.489	0.548
Attendance	0.441	0.289

Table 4 shows that laboratory exercises have the maximum gain ratio and it is selected as the splitting attribute what can be seen from Figure 4.

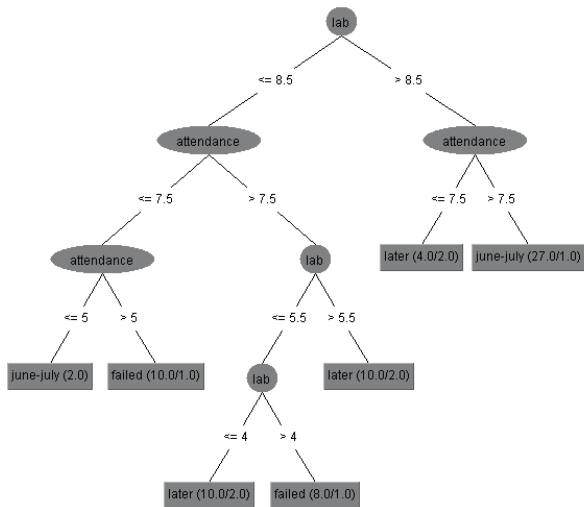
In this case, the achieved accuracy is 61.9% and created decision tree is shown in Figure 5 a. Increasing the minimum number of instances per node (minNumObj) from 2 to 4, the accuracy drops to 57.5% creating simpler tree shown in Figure 4 b. Applying decision tree classifier on balanced data the achieved accuracy is 73.2%. Tree created on balanced data set is shown in Figure 5 c.



a)



b)



c)

FIGURE 5 DECISION TREE (3RD EXPERIMENT): A) THE INITIAL MODEL, B) THE MODEL ACHIEVED BY INCREASING THE MINIMUM NUMBER OF INSTANCES PER NODE, C) MODEL ACHIEVED OVER A BALANCED DATA

4th experiment

In the fourth experiment the influence of individual laboratory exercises (L1, L2 and L3) and the student attendance on lectures to passing the exam in the first, June-July, final exam term is analyzed.

Results of attributes evaluation in this experiment are shown in Table 5.

TABLE 5. ATTRIBUTES EVALUATION – 4TH EXPERIMENT

Attribute	InfoGain	GainRatio
	AttributeEval	AttributeEval
L1	0.212	0.214
L2	0.384	0.335
L3	0.363	0.321
Attendance	0	0

Table 5 shows that the attribute with the maximum gain ratio is L2 and it is selected as the splitting attribute while evaluation of attribute attendance is 0.

In this case the J48 decision tree classifier achieves an accuracy of 47.8% and created decision tree is shown in Figure 6 a. From confusion matrix (Table 6) it can be seen that there is an imbalance in the distribution of the value of output classes and the accuracy of small classes is less than the accuracy of the higher class.

TABELA 6. CONFUSION MATRIX – 4TH EXPERIMENT

Predicted class			Real class
a	b	c	
25	4	6	<i>a=june-july</i>
7	4	5	<i>b=failed</i>
10	5	5	<i>c=later</i>

Applying the function Resample data distribution balance is improved, which affects the result. In this case the accuracy of 80.2% is achieved. Thus, the predictive accuracy of a balanced data is significantly increased. If in the balanced data the number of instances per node is increased from 2 to 3, the accuracy of model predictions fails to 77.4% and the decision tree classifier creates a simpler decision tree Figure 6 b.

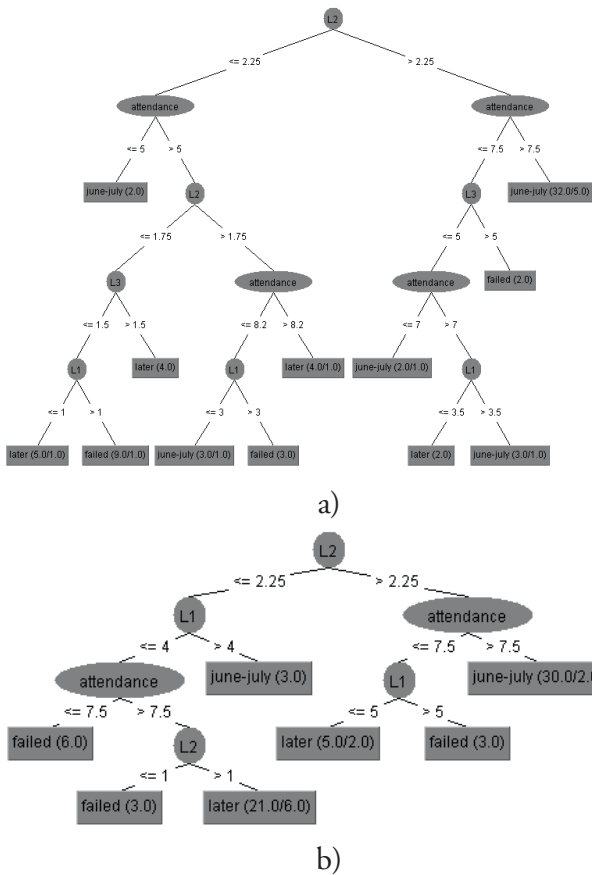


FIGURE 6 DECISION TREE (4TH EXPERIMENT): A) THE INITIAL MODEL, B) THE MODEL ACHIEVED BY INCREASING THE MINIMUM NUMBER OF INSTANCES PER NODE OVER A BALANCED DATA

Analyzing the summarized results of performed experiments, presented in Table 7, it can be seen that the highest accuracy of the initial data is achieved in the second experiment, while the lowest accuracy is achieved in the fourth.

TABLE 7 SUMMARIZED RESULTS OF PERFORMED EXPERIMENTS

experiment/achieved accuracy	initial model	balanced data
First experiment	71.8%	76.5%
Second experiment	74.6%	87.3%
Third experiment	61.9%	73.2%
Fourth experiment	47.8%	80.2%

Also, it is evident that with more class attributes accuracy decreases. By increasing the minimum number of instances per node a decision tree is simpler but at the same time accuracy decreases.

After balancing the data, accuracy in all four experiments is significantly increased, with the highest

accuracy achieved again in the second experiment, while the largest percentage improvement over the initial model is discernible in the fourth experiment. Experiments have shown that the greatest influence on the outcome of students success in the final exam has the laboratory exercise L2 while L1 has the smallest influence.

CONCLUSION

After all discovered patterns or gained knowledge by applying data mining algorithms it is necessary to discover those that are useful for the particular application and to identify variables that can help teachers in predicting student performance. Experiments performed in this work using the J48 decision tree classifier showed that the laboratory exercise of the second cycle have the greatest impact on the success of passing the exam which leads to a conclusion that this teaching unit need an extra attention. Also, improving its content should lead to better overcome of those laboratory exercises and thus directly influence increase of the final exam passing rate.

From filter method and the obtained experimental results it can be concluded that the impact on the learning process have only those activities in the course that are mandatory. This imposes a recommendation that a greater number of activities should be classified into this category in order to ensure better and continuous work of the students during the semester, what will be the subject of our future research.

Authorship statement

Author(s) confirms that the above named article is an original work, did not previously published or is currently under consideration for any other publication.

Conflicts of interest

We declare that we have no conflicts of interest.

LITERATURE

- [1] Affendey, L.S., et al. (2010). Ranking of Influencing Factors in Predicting Students' Academic Performance, *Information Technology Journal* 9 (4): 832-837.
- [2] Ai, J., and Laffey, J. (2007). Web Mining as a Tool for Understanding Online Learning, *MERLOT Journal of Online Learning and Teaching* 3(2): 160-169.
- [3] Kickul, J., and Kickul, G. (2002). New pathways in e-learning: The role of student proactivity and technology utilization, 45th Annual Meeting of the Midwest Academy of Management Conference, Indiana, USA.
- [4] Liu, B. (2007). Web DataMining - Exploring Hyperlinks, Contents, and Usage Data, © Springer-Verlag Berlin Heidelberg
- [5] Moodle, Available: <https://moodle.org/>
- [6] Prema, M., and Prakasam, S. (2013). Effectiveness of Data Mining - based E-learning system (DMBELS), *International Journal of Computer Applications* 66(19): 31-36.
- [7] Ramesh, V., et al. (2013). Predicting Student Performance: A Statistical and Data Mining Approach, *International Journal of Computer Applications* 63(8): 35-39.
- [8] Romero, C., and Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005, *Expert Systems with Applications* 33, 135-146.
- [9] Romero, C., et al. (2008). Data mining in course management systems: Moodle case study and tutorial, *Computers&Education*, Elsevier 55(1):368-384.
- [10] Romero, C., and Ventura, S. (2013). Data mining in education, *WIREs Data Mining Knowl Discov*, 3(1): 12-27.
- [11] Romero, C., et al. (2013). Web usage mining for predicting final marks of students that use Moodle courses, *Comput. Appl. Eng. Educ.*, 21: 135-146.
- [12] Weka software tool, Available: <http://www.cs.waikato.ac.nz/ml/weka/>
- [13] Witten IH et al. (2011) Data mining: practical machine learning tools and techniques, Morgan Kaufmann, Amsterdam

Submitted: November 18, 2013.

Accepted: December 20, 2013.

OBJECT-ORIENTED ANALYSIS AND DESIGN FOR ONE ALGORITHM OF COMPUTATIONAL GEOMETRY: FORWARD, REVERSE AND ROUND-TRIP ENGINEERING

Muzafer H. Saračević, Predrag S. Stanimirović, Sead H. Mašović

Department of Computer Science, Faculty of Science and Mathematics,

University of Nis, Visegradska 33, 18000 Nis, Serbia.

muzafers@uninp.edu.rs, peckois@pmf.edu.rs, sead.masovic@pmf.edu.rs

Case study

DOI: 10.7251/JIT1302096S

UDC: 378.046.4:004.41]:004.428

Abstract: *Triangulation of the polygon is a fundamental algorithm in computational geometry. This paper considers techniques of object-oriented analysis and design as a new tool for solving and analyzing convex polygon triangulation. The triangulation is analyzed from three aspects: forward, reverse and round-trip engineering. We give a suggestion for improving the obtained software solution of the polygon triangulation algorithm using technique that combines UML modeling and Java programming.*

Keywords: *Software engineering, Computational geometry, Triangulation of Polygons, Modeling in UML, Java.*

INTRODUCTION

Triangulation enables to get a display of three-dimensional objects from a set of given points and provides a mechanism for so-called glazing of three-dimensional figures [8]. Polygon triangulation has many applications in computer graphics and it is used in the pre-trial phase of non-trivial operations of simple polygons [10]. Triangulation of convex polygons is an actual problem which appears in the two-dimensional computational geometry [14,19]. Triangulation of a convex polygon assumes decomposition of the polygon interior into triangles by internal diagonals that are not intersected.

Polygon triangulation is a complex problem that requires complex class for an efficient object-oriented implementation. In order that this class would be comprehensible, it is necessary to do their analysis. Dealing with the complexity of the same, there is a need for new techniques to develop alternative views and engineered for the field of object-oriented modelling.

This paper presents an object-oriented analysis and design (OOAD) based on Hurtado-Noy method for the triangulation of convex polygon, which is introduced in [11]. OOAD provides a comprehensive insight into the implementation of this problem.

We present analysis and design for the Hurtado-Noy method through three aspects, which can be briefly described as follows:

1. Direct development (*forward engineering*): this approach is based on generating the source code in a selected programming language from the *UML* model (Unified Modeling Language). In our case, we have chosen the programming language *Java*.
2. Feedback analysis (*reverse engineering*): it refers to the interpretation of the source code that is generated from defined *UML* models in a selected programming language.
3. Synchronization of feedback analysis and direct development (*round-trip engineering*):

this aspect investigates the synchronization between the source code changes and *UML* models.

BASELINES AND PRELIMINARIES

The number of all triangulations of a convex polygon with n vertices is equal to the $(n-2)$ th Catalan number

$$C_{n-2} = \frac{(2n-4)!}{(n-1)!(n-2)!}, n \geq 3 \tag{1}$$

For more details about the convex polygon triangulation see for example [15].

Details of *Hurtado-Noy* method [11]: Let $T(n)$ be the set of triangulations of an n -gon. Every triangulation t that belongs to $T(n)$ has exactly one “predecessor” in $T(n-1)$ and one or more “descendants” in the set of triangulations $T(n+1)$. For a given set $T(n)$, there is a possibility to generate triangulations of the $(n+1)$ -gon derived from arbitrary triangulation $t \in T(n)$. This principle is illustrated in Figure 1.

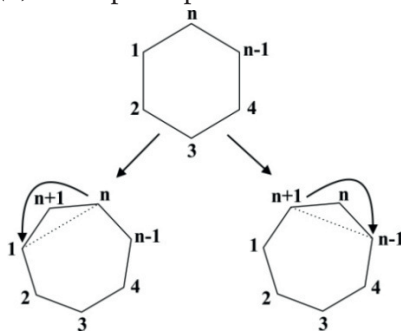


FIGURE 1. FORMING THE NEW TRIANGULATION FOR $(N+1)$ -GON, ACCORDING TO *HURTADO-NOY* METHOD

Algorithm 1 describes the *Hurtado-Noy* method from [11].

Algorithm 1. *Hurtado-Noy* method

Require: Positive integer n

- 1: Check the structure containing $2n-5$ vertex pairs looking for pairs $(i_k, n-1)$, $i_k \in \{1, 2, \dots, n-2\}$, $2 \leq k \leq n-2$, i.e. diagonals incident to vertex $n-1$. The positions of these indices i_k within the structure describing a triangulation should be stored in the array.
- 2: For every i_k perform the transformation $(i_l, n-1) \rightarrow (i_l, n)$; $i_l < i_k$, $0 \leq l \leq n-3$.
- 3: Insert new pairs (i_k, n) and $(n-1, n)$ into the structure.
- 4: Take next i_k , if any, and go to Step (2).
- 5: Continue the above procedure with next $(n-1)$ -gon triangulation (i.e. structure with $2n-5$ vertex pairs) if any. Otherwise halt.

Based on the above principle of separation of predecessor, *Hurtado* and *Noy* provided the hierarchy which is important because of its inherent simplicity and also owing to the fact that it has a number of really exciting properties (see Figure 2, restated from [11]).

An implementation of this algorithm in *Java* programming language is presented in our paper [23]. In the mentioned implementation, the phase of coding on the basis of a given Algorithm 1 was performed without prior analyzing and creating a visual plan. This way of solving the problem can adversely affect the functionality and intelligibility of generated source code.

A better understanding and detailed analysis of *Hurtado-Noy* method is achieved applying forward engineering on the same algorithm. In addition, we get a developed visual model (plan for solving) which is independent of implementation and technology. This approach deals with the defined model that allows the transition to the phase of coding (programming). After finishing the programming phase, reverse and round-trip engineering play a key role in the maintenance and evolution of the obtained solution for Algorithm 1.

This paper is organized as follows. Section 2 describes the *UML* modeling process appropriate for generating triangulations of the convex n -gon. This section presents the possibility of generating *Java* source code from *UML* model. Also, in this section we presented *Java* experimental results obtained by the developed software solution. A method for improving already created software solutions in a selected programming language is given in the section 3. The improvement is based on advanced techniques

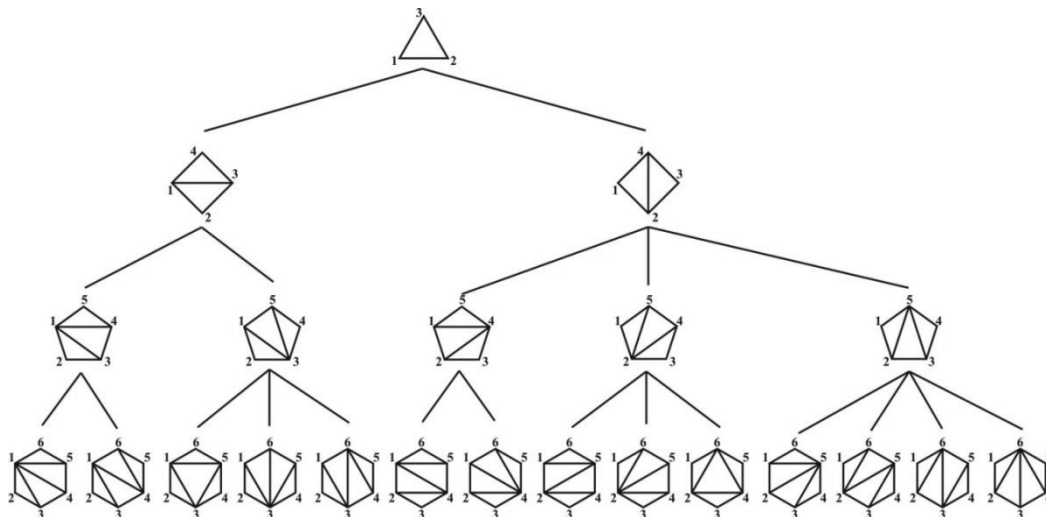


FIGURE 2. LEVELS THREE TO SIX OF THE TREE OF TRIANGULATIONS - HURTADO NOY HIERARCHY

for reverse engineering and synchronization of the UML models and the Java source code. Advantages of all three approaches are listed in the last section.

RELATED WORKS

UML modelling has found various applications that cover a wide spectrum of different application domains. During software evolution, programmers devote most of their effort to the understanding of the structure and behaviour of the system.

The paper [20] proposes an UML-based software maintenance process. The authors give the descriptions as variants of UML profiles, describing the styles and rules relevant for a particular application domain. A reverse engineering sub-process, combining top-down and bottom-up reverse engineering activities, aims at constructing the architectural models. The authors describe some of the most advanced techniques that can be employed to reverse engineer several design views from the source code. The paper [18] presents the form driven object-oriented reverse engineering methodology by using forms to recover semantics of legacy applications. The authors propose the application to demonstrate the practical usability of the object-oriented reverse engineering methodology by transforming the resulting object models into well-known UML-based models [12].

The paper [7] presents code reverse engineering problem for identification object-oriented source codes. Paper [21] proposes reverse engineering se-

quence diagrams from enterprise Java beans with interceptors. In the paper [2] authors present an approach and tool to automatically instrument dynamic web applications using source transformation technology and to reverse engineer an UML sequence diagram from the execution traces generated by the resulting instrumentation. The authors of the paper [24] propose combination of the three relations in such way that enables a comprehensive measure of complexity of class diagrams in reverse engineering. A research that is relevant to the application of UML use case diagrams and their comparison, in order to obtain the best possible software at the stage of verification and validation of the software, is presented in [9].

The paper [4] describes the procedure of modelling at the level of hardware, systems and algorithms. The article [16] describes the interaction between behaviour diagrams (activities and states) and interaction diagrams. The method of automatic consistency checking between the two given diagrams is given in [5]. The paper [13] represents the solution of the object-oriented approach in the design and implementation of web based solutions through UML and Java code. Full analysis of how to model one problem in educational purposes and represent it in a comprehensible way is given in the paper [3].

FORWARD ENGINEERING FOR POLYGON TRIANGULATION ALGORITHM

UML is a language that creates an abstract model of the system through a set of graphical diagrams. It

can be used for specification, visualization, designing and documentation of the systems development. General classification of standard *UML* views (models) can be divided into: static, dynamic and physical model. Modeling in *UML* has various applications [1,12,20,25] that cover a wide spectrum of different application domains. We monitor the implementation of the convex polygon triangulation by means of *UML* modeling. This monitoring is carried out from abstract ideas, through particular classes, activities, states and behavior of the system, until the physical distribution.

Modeling in UML: static, dynamic and physical model

Our project for modeling Hurtado-Noy algorithm contains 47 diagrams totally, 36 of which are associated with the dynamic model. Table 1 presents all listed diagrams.

TABLE 1. OVERVIEW OF *UML* DIAGRAMS

Type	UML diagrams	No. of diagrams
Static models	Class diagrams	1
	Object diagrams	8
	Use case diagrams	8
	Activity diagrams	7
Dynamic models	State-Chart diagrams	7
	Sequence diagrams	6
	Collaboration diagrams	4
	Interaction overview diagram	4
Physics models	Deployment diagram	1
	Component diagram	1

All models of the system which are obtained from the developed environment (*Visual Paradigm for UML* and *NetBeansUML*), can be downloaded from [27].

Class diagram describes the static structure of the system. Classes are modeled and mutually connected using these diagrams, while the objects are described by their attributes and relationships with other objects. Each object has a number of methods that can be performed, which is modeled by his behavior. The *UML* class diagram for *Hurtado-Noy* algorithm is presented on Figure 3. These classes are called *Gen-*

erateTriangulations, *Triangulation*, *Node*, *LeafNode*, *Point* and *PostScriptWriter*.

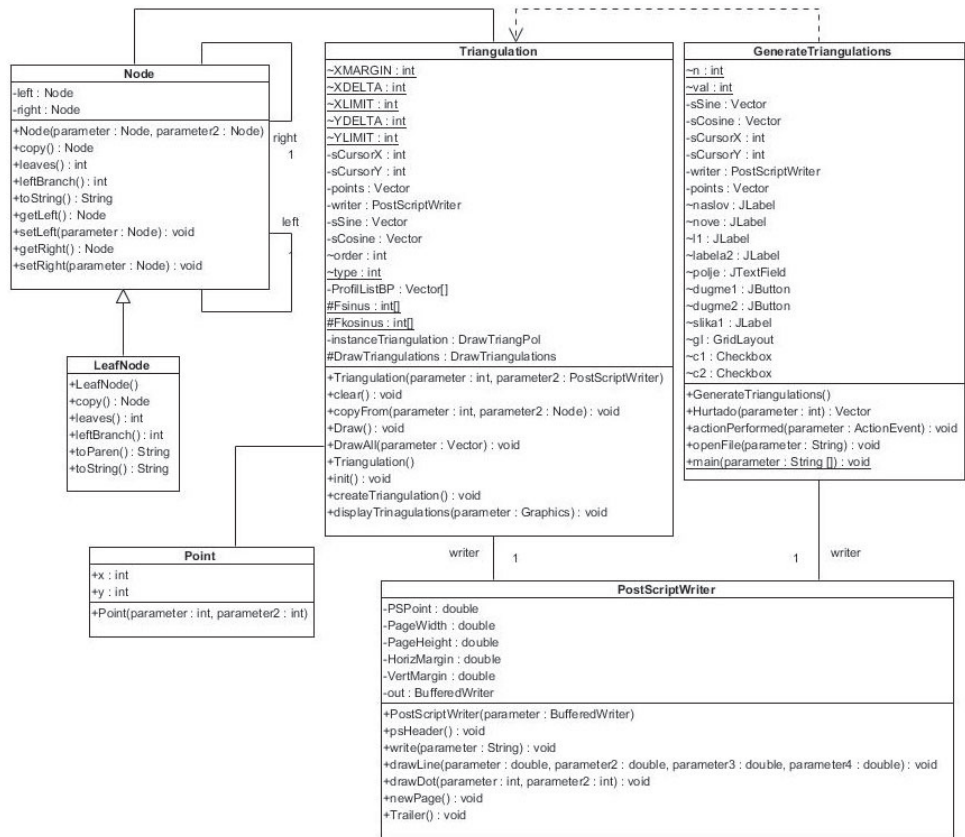
The class *Triangulation* is responsible for displaying a convex polygon triangulation. This class provides verification of all the vertices of the polygon. Method *Draw* is a member of the class *Triangulation*. This method is responsible for making an individual triangulation, which is defined by the underlying combination of internal diagonals. Method *DrawAll* also belongs to the class *Triangulation* and it provides iteration where the method *Draw* is called C_{n-2} times. The *Hurtado* method (member of the class *GenerateTriangulations*), creates string of objects of the class *Node* that is used to obtain the appropriate number of vertices (nodes) to form the regular convex polygon. The method *drawLine* is responsible for drawing regular convex polygons. The main executive method in *Java* is defined in the class *GenerateTriangulations*, that requires the input parameter .

Three types of connections are defined in the *Class Diagram*: dependency, generalization and association. Dependency is most commonly used when one class uses another one as an argument. Example of the dependency connection is the relation between the class *Triangulation* and the main class *GenerateTriangulations* (see Figure 3). The Association is a relationship which specifies that objects are associated with other objects (e.g. *Triangulation* with *Node* and *Triangulation* with *Point*). Generalization is a relationship between classes where one class shares the structure and/or behavior of one or more classes. Generalization also defines a hierarchy in which a subclass inherits one or more of the superclass (e.g. *Node* with *LeafNode*).

Operations from the *Class diagram* are further described with behavior and interaction diagrams that together form a dynamic model of the system. Behavior diagrams include activities and state chart views of the system. Interaction diagrams provide data flow between the objects through sequence and communication diagrams.

- Activity diagrams implement the following methods: *createTriangulation*, *Hurtado*, *DisplayTriangulations*, *Draw*, *DrawAll* and etc. (all methods from the Class diagram).

FIGURE 3. CLASS DIAGRAM



- State-Chart diagrams are used to give an abstract description of the algorithm's behavior. An example of the transition from one state to another is a process of generating triangulation using appropriate methods for moving into their recording and drawing states. Each of these transition states has three optional parts: alerted event (starting methods for *GenerateTriangulations*), security criteria (aimed to generate an exact number of triangulations, equal to the Catalan number) and activity (drawing triangulation with their notations).
- A sequence diagram shows object (class) interactions arranged in time sequence and gives a clear display of cooperation between the class *Triangulation* and the main class *GenerateTriangulations*.
- Diagram of collaboration refers to the interaction of objects (all classes in Figure 3).

Use Case Diagram represents the functional requirements which are imposed to the system. Within the problem of triangulation of polygons, we can observe the following use cases illustrated on Figure 4: *generating triangulation*, *storage (recording)* and *drawing triangulation*.

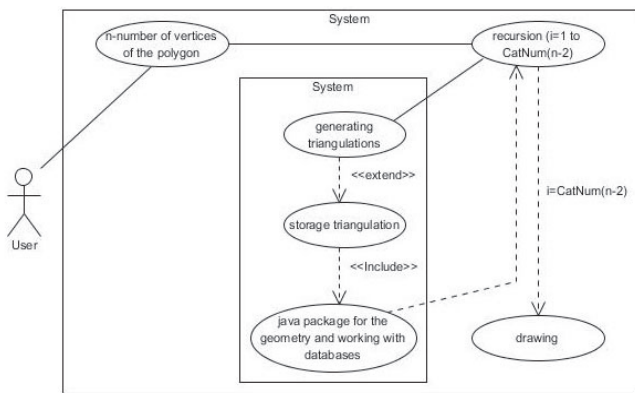


FIGURE 4. USE CASE DIAGRAM

Figure 4 illustrates the general division of all methods into three groups:

- methods that generate triangulation (with drawing and storing it in the output file),
- methods responsible for the assignment of appropriate notation,
- methods that store the triangulation in different formats through JDBC API.

In addition to these activities, the last stage is drawing triangulation. This stage is supported by corresponding Java package for the geometry [22].

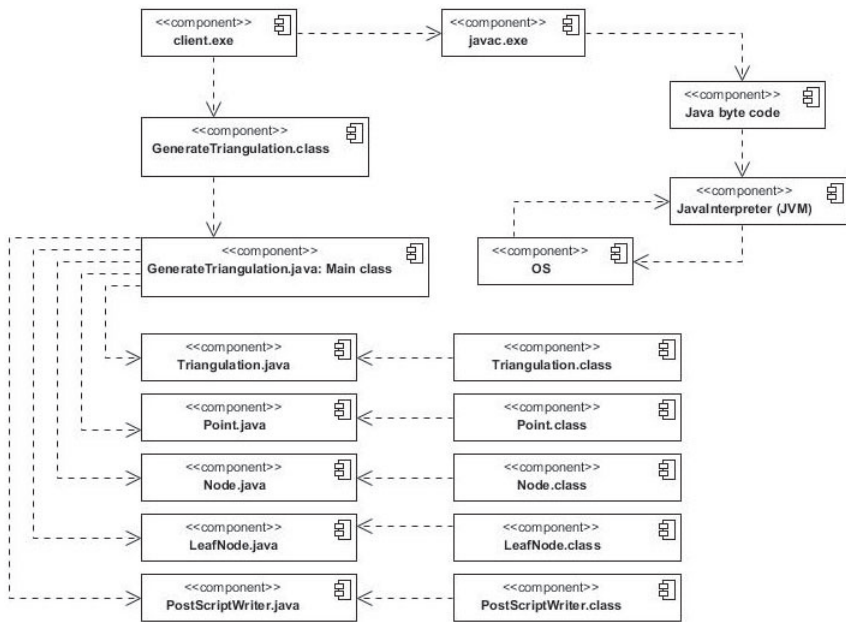


FIGURE 5. COMPONENT DIAGRAM

The physical model is implemented through the component diagrams and development (or deployment) diagrams. Deployment diagram shows the hardware structure of the system, i.e. the communication between hardware and software components (dependency connection *supports*). Software component is the implementation of methods, while hardware component is *Java* JDK platform with its components to support the implementation.

The component diagram (Figure 5) represents the structure of the system and describes the dependence of the components of the system. The elements of the diagram are the source codes, library, dynamic components and executable programs.

Generating Java source code from UML models

The idea of generating source code is always associated with the tools and techniques that are based on *UML*. The source code based on the class diagram could be generated in some environments, such as *Visual Paradigm for UML* and *NetBeansUML*. The process of generating source code based on the created model leads to the general structure of a software solution for the algorithm. In this way, we can provide a fast and efficient transfer model in a customizable *Java* source code. The application of the direct development on the *Class Diagram* we get all the classes with the general structure (i.e. declarations of variables and headers of their methods).

The complete structure that is obtained from the UML models can be downloaded from [28].

Example 1. Here we specify one example of generating one segment of the *Java* source code for the classes *Triangulation* and *GenerateTriangulations*. The sign “*” denote that there is a possibility for the code modification in order to achieve the necessary and/or desired functionalities.

```

public class Triangulation{
//attributes public
Object private int sCursorX;
public Object private int sCursorY;
public Object private Vector<Point>
points;
public Object private PostScriptWriter
writer;

***

//operations
public void Draw() {*}
public void DrawAll(Vector<Node> trees)
() {*}
public void clear() {*}
public void copyFrom(Object int aOffset,
Object Node t) {*}
}

public class GenerateTriangulation
implements Triangulation
//operations
    
```

```
public void Vector<Node>
createTriangulations (Object int limit)
{*}
public void static void main(Object
String args[]){*}
}
```

Experimental results

Programming phase is the next step that comes after the procedure of direct development. We used the *NetBeans IDE environment* available in *Java* for the implementation of this phase. A comparative analysis of the implementations of the Hurtado-Noy algorithm in three programming languages (*Java*, *Python* and *C++*) is presented in our paper [23]. Numerical experience from this paper shows that the implementation in *Java* programming language produces the best results.

Table 2 contains CPU times required for generating all possible triangulations of convex polygon (denotes the number of the polygon vertices).

TABLE 2. EXECUTION TIME FOR THE JAVA APPLICATION

Num. of vertices	No. of triangulations	Execution time (in sec.)	File output size (in Kb)
5	5	0.2	0.1
6	14	0.3	0.4
7	42	0.4	1.7
8	132	0.5	6.4
9	429	0.6	24.5
10	1430	0.9	93.1
11	4,862	2.2	355.6
12	16,796	5.8	1,363.2
13	58,786	15.2	4,121.4
14	208,012	46.34	11,523.6
15	742,900	124.18	29,874.29

Numerical results are derived using personal computer with performances: *CPU - Intel(R) Core2Duo, T7700, 2.40GHz, L2 Cache 4 MB (On-Die,ATC),RAM Memory -2 Gb, Graphic card - NVIDIA GeForce 8600M GS.*

Based on the obtained results, it can be observed that increasing the values of *n* (number of vertices) increases the number of generated triangulations per second (Figure 6). The vertical axis of the graphical

representation contains the number of displayed triangulations per second while the horizontal axis contains values for *n*.

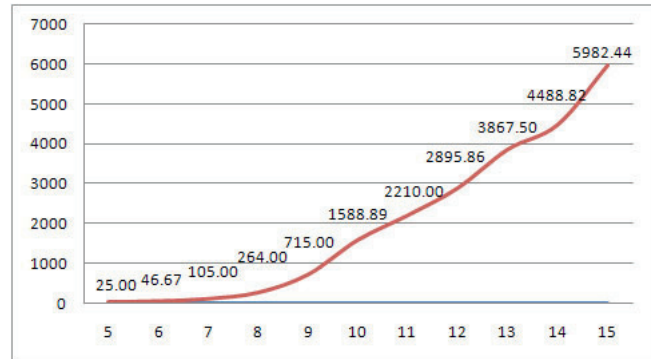


FIGURE 6. NUMBER OF GENERATED TRIANGULATION PER SECOND FOR *N = 5, ..., 15.*

Java application can be downloaded from [29].

REVERSE AND ROUND-TRIP ENGINEERING: MAINTENANCE AND EVOLUTION OF SOFTWARE SOLUTION

This section provides a procedure for the feedback analysis and the ability to synchronize the implementation of algorithm for triangulation, which turned out to be the best solution.

Reverse engineering and visualization of source code lead to improved program comprehension. The main advantages are: *learning unfamiliar code, code reuse, software maintenance, change impact analysis, integrating open source code* and etc. This approach has various applications for identification of the object-oriented codes [6,26].

The reverse engineering for the implementation of a polygon triangulation is implemented through two phases:

1. It begins with the classification of the complete source code in formal units (classes) to obtain the static model (*use-case* and *class diagrams*).
2. On the basis of modeled attributes and operations, their descriptions are further decomposed into dynamic diagrams.

Round-trip engineering presents synchronization of direct development and feedback analysis, which is a good practice in the analysis and maintenance of the implementation [7,17]. Their benefits are di-

rectly related to the change of the source code from *UML* model and vice versa (see Figure 7).

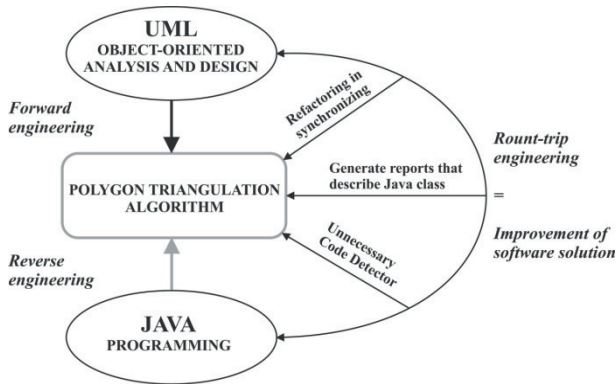


FIGURE 7. SYNCHRONIZATION OF DIRECT DEVELOPMENT AND FEEDBACK ANALYSIS

Example 2. For synchronizing UML project and Java project for the triangulation polygon, NetBeansUML module will log various lines of text to the Output Window as follows:

“...Initial reverse engineering into a new project: Begin processing Reverse Engineering, Parsing 56 elements, Analyzing attributes and operations for 72 symbols, Resolving 54 attribute types, Integrating 72 elements, Building the query cache...”

The output result describes the operations that took place: 72 model elements were used to generate Java source code files. Table 3 presents the fulfillment requirements in the process of synchronizing UML project and Java project for triangulation of a convex polygon (*DP - Design pattern, A - Attributes, O - Operations, I - Implementation, R - Relationships).

TABLE 3. REQUIREMENTS IN THE ROUND-TRIP ENGINEERING

REQUIREMENTS	Triangulation	Generate Triangulations
1 Navigate to Source	+	+
2 Generate Dependency diagram	+	+
3 Generate Code	+	+
4 Generating report	+	+
5 Element Navigation	+	+
6 Refactoring in synchronizing	+	+
7 Find and Replace in <i>UML</i> model	+	
8 Apply DP and source code readability	+	
9 Manipulation with A, O, I and R*	+	+

Description of Requirements: In the *NetBeansUML* module there is the possibility of automatic detecting source code if the synchronization between the *UML* project and *Java NetBeans* project is properly set up (requirements 1,2,3 and 5 from Table 3.1). In this way, it reduces the complexity of the triangulation problem.

Programming and adding new functionality of the system is also facilitated. For the reverse and round-trip engineering process it is important to mention the procedure for generating a report of the model. The report describes all classes defined in the project and the use of packages, interfaces and data types in the implementation (requirement 4).

The main categories of reverse engineering are automatic restructuring and automatic transformation. The first category refers to re-factoring and re-modularization that is applied with the aim of obtaining a better source code (requirements 6 and 7). The second category refers to the application of standards in coding, which is applicable in order to obtain the source code readability (requirement 8).

Re-factoring changes the internal structure of the software (requirement 9) in order to be easier to understand and simpler to modify, without visible changes of his behavior.

The following actions are implemented in our project in the procedure of code re-factoring: *extract and move method, class and super-class; extract interface; use super-type where is possible; creating a template method and encapsulate fields.*

Obtained results in the improved software solution

Table 4 presents the results of improving software solution for the Hurtado-Noy algorithm. Data derived after improvements are presented by the sign ‘*’. Three criteria are used in the comparison: the number of source code lines, the size of *Java* file (in bytes) and measuring the time (speed) in seconds (individually for each *Java* file), for $n = 5, \dots, 15$, cumulative.

TABLE 4. IMPROVEMENT OF SOURCE CODE FOR HURTADO-NOY ALGORITHM

Segment	line	line*	bytes	bytes*	speed	speed*
Triangulation	79	67	2275	1715	29.12	26.14
GenerateTriangulation	222	195	8544	7194	37.56	34.33
Node	45	41	745	578	4.51	4.01
LeafNode	27	19	342	216	3.21	2.85
Point	10	9	134	112	1.52	1.51
PostScriptWriter	56	55	1830	1791	2.74	2.62
TOTAL	439	386	13870	11606	78.66	71.46
Improvement (%)	13.73		19.51		10.08	

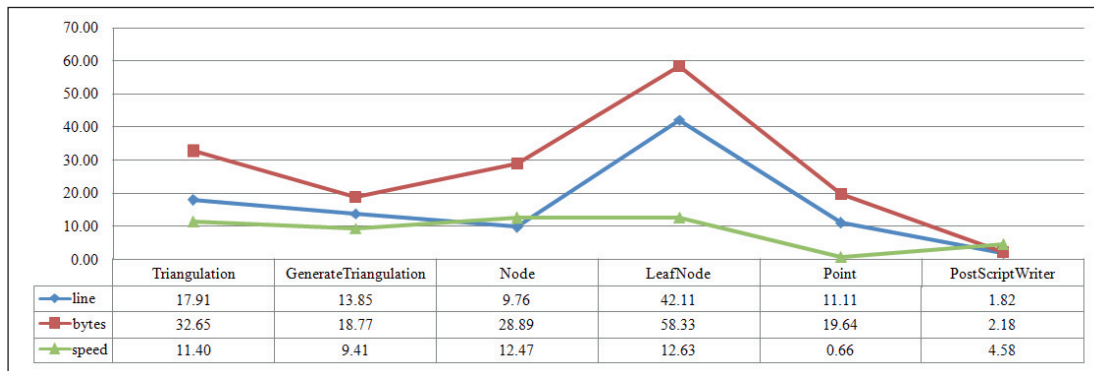


FIGURE 8. IMPROVEMENTS (IN %) FOR HURTADO-NOY ALGORITHM (INDIVIDUALLY BY CLASSES)

Figure 8 shows the percentage of improving the source code for three criteria, individually for each segment of the implementation.

In process of code review for Hurtado-Noy algorithm, *NetBeans module "Unnecessary Code Detector"* recognizes the following: *unused imports, unread local variable, unread parameter, unnecessary method or constructor, unread private method, constructor or type and unread local or private members.*

Some advantages that occur in the application of the reverse and round-trip engineering in our implementation are:

1. Better understanding of defined classes and their methods, identifying interdependencies, ways of communication and data flow. Hence, given technique offers the possibility of generating alternative views of the problem.
2. Source code analysis obtained through several models (primarily static and dynamic) provide the possibility to expand the source code and simplify methods. This allows the detection of repeated cases in the code.

TABLE 5. ADVANTAGES OF THREE APPROACHES

Engineering	Advantages
Forward	Multi-dimensionality of the system and the high level of abstraction
	Efficient transparency of the system structure
	Spotting the functional wholes
	Generating the source code
Reverse	Analysis and interpretation of the implementation problems
	Logical design and better visibility of source code
	Disassemble of <i>Java</i> code
	More effective maintenance of a software solution
Round-Trip	Synchronized changes from model to source code or vice versa
	Generate reports that describe class (graphic and program description)
	Combining the advantages of the first two approaches

3. After locating and removing the source code or modules that are not used anymore, we reduce the complexity of the problem and simplify the source code. Therefore, we achieved better results concerning the speed of generating triangulations per second.

Table 5 shows the identified advantages of all three approaches in the implementation.

CONCLUSION

This paper outlines the key advantages of the object-oriented analysis and design in solving the convex polygon triangulations, which is a fundamental algorithm in computational geometry. Direct development has the advantage of generating the source code in some of the object-oriented programming languages, while reanalysis technique aims to describe the implementation of the problem through various aspects. Synchronization procedure combines the advantages of these two approaches.

Based on the presented testing, we conclude that the best practice is the synchronization technique that combines *Java* programming and *UML* modeling. Some of the advantages of reverse engineering and synchronization of direct development and feedback analysis to solve the problem of triangulation are coping with the complexity of the problem, better understanding of the classes and their methods, identifying interdependencies, ways of communication and data flow. In addition, a given technique offers the possibility of generating alternative views of

the problem. Source code analysis obtained through several models allows you to see the problem from the aspect of expandability of the source code (adding new methods), possibilities of simplification of methods, redefining the methods and synthesis and analysis methods. In this way they can link certain implementation on the basis of their dynamic and static models. This allows the detection of secondary occurrences and repeated cases. OOAD technique enables reuse of already implemented classes, in terms of easy and efficient adding new attributes and operations.

Obtained results indicate improvement of software solutions through three aspects: the number of source code lines, the size of output file and speed of execution. The archival value of the paper is a contribution to the engineering education through a case study in the computational geometry. This technique of three approaches can be applied as a new method for solving and analyzing related problems. Generally, the suggested approach is suitable for the implementation of some other algorithms in computational geometry.

Acknowledgement

The authors gratefully acknowledge support from the Research Project 174013 of the Serbian Ministry of Science.

Authorship statement

Author(s) confirms that the above named article is an original work, did not previously published or is currently under consideration for any other publication.

Conflicts of interest

We declare that we have no conflicts of interest.

REFERENCES

- [1] Aghasaryan, A., Jard, C., and Thomas, J. (2004). UML Specification of a Generic Model for Fault Diagnosis of Telecommunication Networks. In Proceedings of 11th International Conference on Telecommunications, Fortaleza, Brasil, 841-847.
- [2] Alalfi, M.H., Cordy, J.R., and Dean, T.R. (2009). Automated Reverse Engineering of UML Sequence Diagrams for Dynamic Web Applications. In IEEE international conference on software testing, verification, and validation workshops, 287-294.
- [3] Ayachi-Ghanouchi, S., Cheniti-Belcadhi, L., and Lewis, R. (2013). Analysis and modeling of tutor functions. *Computer Applications in Engineering Education*, 21 (4), 657-670.
- [4] Bahill Tand Daniels, J. (2002). Using object-oriented and UML tools for hardware design: A case study. *Systems Engineering*, 6 (1), 28-48.
- [5] Belt, J. (2005). Automated Consistency Checking between UML State Charts and Sequence Diagram. CIS 798.
- [6] Bruegge, B. (2004). Object-Oriented software engineering: Using UML, patterns and Java. Pearson Education, New Jersey.
- [7] Bringer, J., and Chabanne, H. (2012). Code Reverse Engineering Problem for Identification Codes. *IEEE transactions on information theory*, 58(4), 2406-2412.
- [8] Chen Jand Chen, C. (2008). Foundations of 3D Graphics Programming: Using JOGL and Java3D. Springer, New York.

- [9] Funkhouser, O., Etzkorn, L., and Hughes, W. (2008). A Lightweight Approach to Software Validation By Comparing UML Use Cases with Internal Program Documentation Selected Via Call Graphs. *Software Quality Journal*, 16 (1), 131-156.
- [10] Garey, M.R., Johnson, D.S., Preparata, F., and Pand Tarjan, R.E. (1978). Triangulating a simple polygon. *Inform. Process. Lett.*, 7, 175-180.
- [11] Hurtado, F., and Noy, M. (1999). Graph of Triangulations of a Convex Polygon and tree of triangulations. *Computational Geometry*, 13, 179-188.
- [12] Huynh, S., Cai, Y., and Shen, W. (2008). Automatic Transformation of UML Models into Analytical Decision Models. Technical Report DU-CS-08-01, Drexel University.
- [13] Jayaramaraja, S. (2005). An object-oriented design and reference implementation for web-based instructional software. *Computer Applications in Engineering Education*, 13 (1), 26-39.
- [14] Klawonn, F. (2012). Introduction to Computer Graphics: Using Java 2D and 3D: Second Edition. Springer, New York.
- [15] Koshy, T. (2009). Catalan Numbers with Applications. Oxford University Press, New York.
- [16] Knapp, A., and Merz, S. (2002). Model Checking and Code Generation for UML State Machines and Collaborations Tools for System Design and Verification. Institut fur Informatik, Universitat Augsburg, 59-64.
- [17] Lano, K. (2005). Advanced Systems Design with Java, UML, and MDA. Elsevier publisher.
- [18] Lee, H., Yoo, C. (2000). A form driven object-oriented reverse engineering methodology. *Information systems*, 25 (3), pp. 235-259.
- [19] Loera Jand Santo, F. (2003). Triangulations: Structures for Algorithms and Applications. Springer Verlag, New York.
- [20] Riva, C., Selonen, P., Systa, T., and Xu, J. (2004). UML-based reverse engineering and model analysis approaches for software architecture maintenance. In Proceedings of 20th IEEE international conference on software maintenance, USA, 50-59.
- [21] Roubtsov, S., Serebrenik, A., Mazoyer, A., and Brand, M. (2011). I2SD: Reverse Engineering Sequence Diagrams from Enterprise Java Beans with Interceptors. In 11th IEEE international working conference on source code analysis and manipulation (SCAM 2011), 155-164.
- [22] Saračević, M., Stanimirović, P., and Mašović, S. (2013). Implementation of some algorithms in computer graphics in Java. *Technics Technologies Education Management*, 8 (1), 293-300.
- [23] Saračević, M., Stanimirović, P., Mašović Sand Biševac, E. (2012). Implementation of the convex polygon triangulation algorithm. *Facta Universitatis, series: Mathematics and Informatics*, 27 (2), 67-82.
- [24] Sheldon, F.T., and Chung, H. (2006). Measuring the complexity of class diagrams in reverse engineering. *Journal of software maintenance and evolution-research and practice*, 18 (5), 333-350.
- [25] Stanimirović, P., Tasić, M., Saračević, M., and Mašović, S. (2012). UML-based modeling for Moore-Penrose inverse computation. *Revista Metal. International*, 17 (12), 99-106.
- [26] Tonella, P. (2005). Reverse engineering of object oriented code. In Proceedings of 27th International Conference on Software Engineering. Missouri, USA, 724-725.
- [27] Link for all UML models:<http://muzafers.uninp.edu.rs/triangulation/UMLTriang.rar>
- [28] Structure of Java Code:<http://muzafers.uninp.edu.rs/triangulation/GeneralStructureHurtado.rar>
- [29] Link for Java application:<http://muzafers.uninp.edu.rs/triangulation/Hurtado.rar>

Submitted: October 30, 2013.

Accepted: December 7, 2013.

CRM PERFORMANCES ACCENTED WITH THE IMPLEMENTATION OF DATA WAREHOUSING AND DATA MINING TECHNOLOGIES

Ines Isaković

isakovic.ines@hotmail.com

General survey

DOI: 10.7251/JIT1302107I

UDC: 37.018.43:519.816

Abstract: Customer Relationship Management (CRM) has become more and more a key strategy for large and small businesses. It supports marketing, sales, services and involves direct and indirect customer interaction. Customers are put into the center of the business, because they represent an asset and profit for any company. Customers need to be satisfied in order to be loyal. A company can achieve that by meeting customer's needs and expectations. In order to perform both for the benefit of the customer and for itself, a company has to use all the positive advantages of IS technologies that support CRM including data warehouses and data mining, that are clearly presented in this paper

Key Words: Customer Relationship Management (CRM), Data Warehousing (DW), Data Mining (DM)

INTRODUCTION

New information technology systems have an impact on the trend of a global economy redesigning that causes the fast reduction of technological costs. Today's Web sites, e-mails, computers and other automation tools made feasible implementation of the cost efficient Customer Relationship Management (CRM) into the organizations. Because of that, Data Mining and Data Warehousing, as same of automated tools, are crucial for the company, because those tools help to get useful information (reports) about customers and their needs.

The organizations cannot escape this revolution, because it is present and it is *business-to-business (B2B)* marketing revolution. Each level of the company will be affected, but some of the company managers will follow new trends and some of them will refuse changes and the role of the new technology. However, to avoid misunderstanding, the implementation of the Customer Relationship Management (CRM) will not be an easy process, and the fact is that organizations that do not accept those changes will lose a competitive advantage.

DATA WAREHOUSING (DW)

A data warehousing "is a copy of transaction data specifically structured for querying, analysis, and reporting [5]." Data warehousing uses data from any type of database and analyses them, for example customer services such as complaints and compliments and give report, from which, company can easily see most common customers complaints or rewords. **Data warehouse database** "is a database that focuses primarily on the storage of data used to generate the information required to make tactical or strategic decisions [6]."

The unique form of a data warehouse is the **operational data store (ODS)**. This form is much smaller than the conventional data warehouse, because it stores only information about customer identity. The operational data store is structured for transactional performance. This kind of data warehouse is used for the front-office systems and for the purpose of establishing a single view of the customer.

The **data conversion** is constructed in a way that data is copied from tactical databases to the data

warehouse. In this way, the duplication of data is reduced and database inconsistencies are solved.

Selecting and combining technology options for CRM

The integration of the CRM approach will also be determined on the organization’s CRM strategy. In terms of data repository CRM strategy development is based on four broad alternative technologies, which are:

- a database
- data marts
- an enterprise data warehouse, and
- integrated CRM solutions

The graphical illustration of CRM applications is shown in Figure 1.

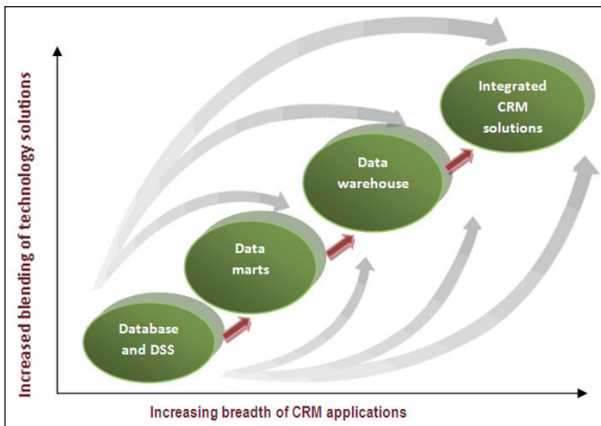


FIGURE 1. TECHNOLOGY LEVELS FOR CRM

(SOURCE: ADRIAN PAYNE, 2005, “HANDBOOK OF CRM: ACHIEVING EXCELLENCE IN CUSTOMER MANAGEMENT”, PAGE 236)

Data mart

Data mart “is the ability of computers to act as an enormous memory and capture all the information on a customer that has been the driving force behind the adoption of CRM IT applications [6].” In order to shift it from the product-based selling to a customer-based marketing, the company needs to have an advance CRM system. A data mart stands for the simplest form of the data warehouse.

Advantages of the Data Mart

Data mart is a tool that will be placed on a department server technology rather than on a PC and will enable numerous of users to connect and use information from it.

It is useful for the businesses that have many departments and would like to respond faster to a business opportunity.

Disadvantages of the Data Mart

The aim of each company that has CRM is to collect needed information about customers and due to that, data warehousing need to be able to store this information. From the beginning, data marts need to be developed as data warehouses. In order to get information based on the best customers and their profitability, product sales and financial data need to be offered. This does not enable companies to develop a consolidated single view of the customer; in that case each department in the business sees the same picture.

Enterprise Data Warehouse

The business that focuses its strategy on the customers will need many databases and data marts. In that case better solution is to have one repository for data. After developing the data warehousing and establishing clean data, analysis on data and data mining software can be applied, and understanding of customers’ manners can start as well as building more advanced CRM strategy.

In addition, data is collected from multiple part of the data warehouse into departments systems. It includes collection of data from databases or department data marts. The data warehouse can track customer relations over the entire customer’s lifetime.

The Figure 2 represents the data warehouse for the CRM systems.

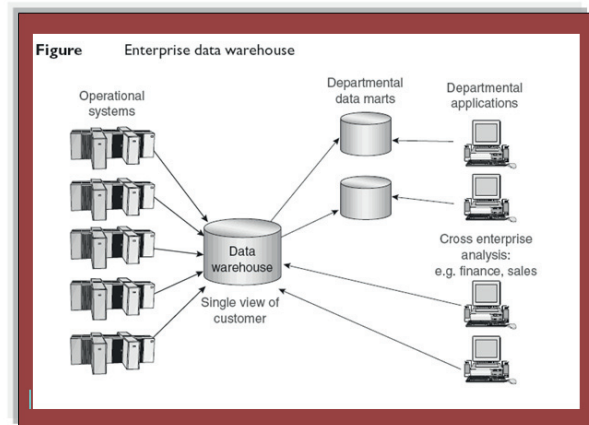


FIGURE 2. ENTERPRISE DATA WAREHOUSE

(SOURCE: ADRIAN PAYNE, 2005, “HANDBOOK OF CRM: ACHIEVING EXCELLENCE IN CUSTOMER MANAGEMENT”, PAGE 242)

Advantages of the Data Warehouse Enterprise

Usage of the data warehouse is beneficial in many cases, such as removing a demand on larger databases. Data that is stored in data warehouse is up to date and periodical (i.e. every 24 hours). In that case, analysis done in different periods will give different results. Also the company can direct analysis in one direction and it can then feed numerous data marts with consistent data.

Disadvantages of the Data Warehouse Enterprise

Enterprise data warehouses are huge and complex IT systems that require continuous implementation for a longer period time. And because of that, businesses are willing to implement cheaper and faster versions for the implementation solutions. Here are some costs and benefits that data warehousing has:

Costs

- Hardware, software, development personnel and consultant costs
- Operational costs such as continuing maintenance of the system

Benefits can be divided into two categories such as:

Added Revenue

- Will the new business objective gather new customers?
- Will the new business objective increase current customer tendency to buy?
- Is the new process necessary to make sure that the competition will not offer a demanded service that you cannot match?

Reduced Costs

- What costs of existing systems will be removed?
- Does the new process have the ability to make some operation more efficient in the future?

DATA MINING (DM)

Data mining, “also known as Knowledge Discovery in Databases (KDD) refers to the efficient process of searching through large volumes of raw data in databases to discover things (e.g. about a customer) that are not easily seen or noticed. The process of searching large amounts of data for patterns is based

on the following methods: clustering, classification and association rules [2].” Data within the data warehousing and data mining can be used to answer some questions related to the organization that a decision maker had not thought to ask before, such as:

- Decision on which products to offer to a current customer?
- Determining the probability that a certain customer will react to a planned promotion?
- Deciding which security will be more profitable to buy or sell during the next trading session?
- What is the probability that a certain customer will try non-payment or pay back on schedule?
- Determining the appropriate medical diagnose for the particular patient?

Data mining can easily help CRM to analyze the largest databases for the purpose of solving business problems. But to make clear, DM is not a business solution, it is a technology same as statistics. On the other hand, CRM transfers information in a database into business decision that establishes a relation with customers. One example where DM helps when we are talking about CRM is deciding on to whom seller needs to send a catalog about the new product. CRM contains a historical database about the earlier connections with customers and all data about the customers. DM uses this data from the historical database and develops a model about the customer behavior that could be used to see which customer will be interested for the new product. This kind of knowledge can be used to send offer to the right customer.

Developing profitable customer relationship using data mining

After developing customer data warehouse, the major issue is how to use all information it contains. As mentioned before, CRM is used in order to increase the profit of the firm via customers' relations. Successful CRM has focused on building a customer database that presents a picture of the customer's relationship with the company.

However, the large amount of customer information as well as ever more complex relations with customers has pushed data mining to the leading po-

sition when we talk about making customer relationships profitable. Data mining is used in discovering patterns by using different methods and a variety of data analysis. Furthermore, it is used to understand what your customers want and predict what they will do. By using data mining, it is easier to determine the right customer, offer the additional products to current customers, such as determining the best customers that can leave the company. CRM applications that use data mining are called analytic CRM.

In CRM, data mining is often used to give a score to a specific customer or prospect where the individual behaves the way we want (i.e. Customer response to a specific product or ability to visit competitors' store and buy products there; segmentation into groups of the customers according to their behaviors, such as buying certain products). Classification can be used in order to determine similar interests held by groups of customers. Another name for classification is collaborative filtering. There are three methods used in data mining:

Classification "builds a classification model by using a given data set, and then classifies them according to their similarities [2]." "An **artificial neural network** is defined as constructing a network of artificial neurons [2]."

Clustering "is dividing data records in a given data set into groups (clusters) according to their similarity [2]."

The association rule "finds a strong association between items using the values of support and confidence, two complex concepts that focus on the similarity between the individual occurrences of the two items, as well as on their co-occurrence [2]."

Three phrases of the customer life cycle:

1. Attracting customers
2. Increasing the value of customers
3. Keeping a good customers

In all three stages of CRM, data mining can help. In addition we will see how.

1. Attracting customers

Building a predictive model, data mining will show who would respond to the company offers (using a decision tree) and using a neural network. In a credit card company, if we know: if customer earns between £20,000 and £40,000 and customer has a house, then the customer risk factor is low. Then, we can issue a credit card to the customer with confidence.

2. Increasing the value of customers

a) Cross-selling via data mining

In order to better understand customers' needs, the company uses data mining methods. When the data mining models are included in a typical cross-selling CRM campaign, then that models help company increase its profit.

b) Personalization via data mining

In order to see which products are grouped into the same group (cluster), the company uses data mining clustering method. After the analysis is done, some clusters are obvious (i.e. in a supermarket, if we know: 98% of customers who purchase orange juice also purchase nappies. Then, we can locate orange juice and nappies together to increase the customer throughput), but some results are unexpected (i.e. A customer who buys books about desert hiking and also buys snakebite kits). These patterns are used in order to group these products together and increase customer purchase.

3. Keeping good customers

For many companies, the costs of attracting new customers go above the costs of keeping good ones. This was the challenge for KnowService that it tries to solve and it consists of three models. One model determines potential customers, the next model picks out the profitable potential customers and the third model matches the potential customers with the most suitable offer. KnowService discovered that the investment given to data mining was beneficial, because it improved customer relationships and increased profitability.

Applying Data Mining to CRM

Some phases have to be included and followed in order to develop a good CRM system. The essential steps of data mining used for successful CRM are:

1. Identify business problem

The essential fact is the need to define business goals, because each CRM application deals with the goals for which, the company needs to develop an appropriate model.

2. Develop database marketing

The essence of this step is data preparation. These two first steps take more time and effort than all the other steps. The data preparation can be done in iterations as well as model building. These data preparation steps may take 50 to 90 percent of the time and effort for the entire data mining process.

The company will need to develop a marketing database, because operational databases and corporate data warehouse do not have the data needed in the form company needs it. Besides, CRM applications could meddle with the quick and effective implementation of these systems.

Clean data, which is needed after the development of the marketing database, is important because of a model developing. That data can be stored in multiple places, and because of that, the company is sometimes in a situation where it needs to integrate and consolidate the data into a single marketing database. The major fact why data quality represents a problem is that the same data are defined in different ways in two databases.

3. Discover data

The first thing that needs to be done before developing a predictive model is to comprehend data that company keeps. The starting point should be collecting numeric summaries, such as descriptive statistics i.e. average, standard deviations etc. Furthermore, continue with the process of analyzing description of the data. In some cases, company wants to develop pivot tables for multidimensional data. In a data preparation phase, the main role plays the tool for creating graphs and data visualization.

4. Prepare data for modeling

In this step, the company finally prepares the data before developing models. This final touch consists of four steps, which are: doing selection on the data that will be used to develop the model. The ideal way

to do this is by putting data into the data mining tool and let the data mining tool find those that are the best predictors.

The next step is to build new predictors resulting from the raw data. After that, the company can decide to choose a subset or sample of data on which to build models. The large amount of available data can cause problems, because it takes more time or in some cases requires buying a better computer.

When samples are selected randomly, it usually results in no loss of information for most CRM problems. So, the company has to change variables according to the requirements of the algorithm based on which it chooses to build the model.

5. Build model

The step in which company constructs the model is an iterative process, in order to find the one that is most useful in solving business problems. What company learns through searching for a good model, may even guide it to go back and make some changes to use the data or modify problem statement. Supervised learning is known protocol on which each CRM application is based. The company starts by using information about customers for which the desired result is known.

6. Evaluate model

Correctness is not important as a good metric for the evaluation of your results. Another measure that is often used is a lift, which measures the improvement attained by a predictive model. This measure does not take into account cost and revenue and due to that it is more suitable and preferable to look at the profit or ROI.

7. Organize model and results

Data mining developed in CRM application is a tiny and dangerous part of the final product. The way data mining is developed in the application is determined by the nature of your customer interaction, which can be done in two ways such that customer contacts company (inbound, i.e. telephone order, Internet order) or company contacts them (outbound, i.e. through advertising or direct mail).

CONCLUSION

While many companies are present in today's marketplace and want to be ahead of these competitors, the implementation of a Customer Relationship Management is crucial in order to compete effectively. Nowadays, CRM is a necessary part of a modern business that puts the customer in the center of its business activities to get long-term satisfaction and loyalty. Today, customers are actually the most important asset of each firm so they should get the full attention of the company's management. Satisfied and loyal customers lead to profitable customers. Customer satisfaction and loyalty are the results of meeting customers' needs and expectations as well as the results of good customer relationships. Companies need to attract new customers and retain current ones by offering good quality products and services. Furthermore, they need to make sure there is a low customer defection rate. A lower defection rate means higher company performance.

In addition to being one step ahead of the competition, companies need to use IS technologies that will support CRM. Through IS technologies companies can get closer to their customers by being able to collect useful information about them as well as being able to satisfy their needs and expectations. Using a data warehouse and data mining, a company can analyze data and get reports with meaningful knowledge that can be further used to improve or change some of their business performances. These technologies are very powerful tools that each leading company should use. All these IS technologies need to be used in a company in order to have a good CRM solution that will support their business strategy.

Authorship statement

Author(s) confirms that the above named article is an original work, did not previously published or is currently under consideration for any other publication.

Conflicts of interest

We declare that we have no conflicts of interest.

REFERENCES:

- [1] Anderson, K. and Kerr, C. (2002) *Customer Relationship Management*. United States of America: McGraw-Hill, [Online]. Available: <http://www.briefcasebooks.com/andersonfm.pdf>. [Accessed April 28, 2008].
- [2] Hongbo Du, (2008) *Lecture Notes*, May 07 2008.
- [3] Kotler, P. (2002) *Marketing Management*, 10 Ed, United States of America: Pearson Custom Publishing [Online]. Available: <http://e-books-for-everyone.blogspot.com/2007/08/marketing-management-millennium-edition.html>. [Accessed April 30, 2008].
- [4] Kotler, P. and Keller, K. L. (2006), *Marketing Management*, 12 ed. United States of America: Pearson Education, Inc.
- [5] Marakas, M. (2003) *Modern Data Warehousing, Mining and Visualization*, United States of America: Pearson Education, Inc.
- [6] Payne, A. (2005) *Handbook of CRM: Achieving Excellence in Customer Management*, Great Britain: Adrian Payne [E-book]. Available: scribd.com. [Accessed May 05, 2008].

Submitted: May 17, 2013.

Accepted: December 2, 2013.

INSTRUCTIONS FOR AUTHORS

The *Journal of Information Technology and Application (JITA)* publishes quality, original papers that contribute to the methodology of IT research as well as good examples of practical applications.

Authors are advised that adherence to the Instructions to Authors will help speed up the refereeing and production stages for most papers.

- Language and presentation
- Length of submissions
- Submission
- Contact details/biographies
- Title of the paper
- Abstract and keywords
- Figures and tables
- Sections
- Footnotes
- Special characters
- Spelling
- References
- Proofs
- PDF offprint
- Copyright and permissions
- Final material
- Correspondence
- Publication ethics

LANGUAGE AND PRESENTATION

Manuscripts should be written in English. All authors should obtain assistance in the editing of their papers for correct spelling and use of English grammar. Manuscripts should have double spacing, with ample margins and pages should be numbered consecutively. The Editors reserve the right to make changes that may clarify or condense papers where this is considered desirable.

LENGTH OF SUBMISSIONS

Papers should not normally exceed 12 Journal pages (about 8000 words). However, in certain circumstances (e.g., review papers) longer papers will be published.

SUBMISSION

Manuscripts must be submitted through the JITA online submission system.

Please read the instructions carefully before submitting your manuscript and ensure the main article files do not contain any author identifiable information.

Although PDF is acceptable for initial submission original source (i.e. MS Word) files will be required for typesetting etc.

CONTACT DETAILS/BIOGRAPHIES

A separate file containing the names and addresses of the authors, and the name and full contact details (full postal address, telephone, fax and e-mail) of the author to whom correspondence is to be directed should be uploaded at the time of submission (you should select Contact details/Biographies as the file type). This file is not shown to reviewers. This file should also contain short biographies for each author (50 words maximum each) which will appear at the end of their paper.

The authors' names and addresses must not appear in the body of the manuscript, to preserve anonymity. Manuscripts containing author details of any kind will be returned for correction.

TITLE OF THE PAPER

The title of the paper should not be longer than 16 words.

ABSTRACT AND KEYWORDS

The first page of the manuscript should contain a summary of not more than 200 words. This should be self-contained and understandable by the general reader outside the context of the full paper. You should also add 3 to 6 keywords.

FIGURES AND TABLES

Figures which contain only textual rather than diagrammatic information should be designated Tables. Figures and tables should be numbered consecutively as they appear in the text. All figures and tables should have a caption.

SECTIONS

Sections and subsections should be clearly differentiated but should not be numbered.

FOOTNOTES

Papers must be written without the use of footnotes.

SPECIAL CHARACTERS

Mathematical expressions and Greek or other symbols should be written clearly with ample spacing. Any unusual characters should be indicated on a separate sheet.

SPELLING

Spelling must be consistent with the Concise Oxford Dictionary.

REFERENCES

References in the text are indicated by the number in square brackets. If a referenced paper has three or more authors the reference should always appear as the first author followed by et al. References are listed alphabetically. All document types, both printed and electronic, are in the same list. References to the same author are listed chronologically, with the oldest on top. Journal titles should not be abbreviated.

Journal

Avramović ZŽ (1995) Method for evaluating the strength of retarding steps on a marshalling yard hump. *European Journal of Operational Research*, 85(1), 504–514.

Book

Walsham G (1993) *Interpreting Information Systems in Organizations*. Wiley, Chichester.

Contributed volume

Huberman AM and Miles MB (1994) Data Management and analysis methods. In *Handbook of Qualitative Research* (Denzin NK and Lincoln YS, Eds), pp 428–444, Sage, Thousand Oaks, California.

Conference Paper

Huberman AM and Miles MB (1994) Data Management and analysis methods. In *Handbook of Qualitative Research* (Denzin NK and Lincoln YS, Eds), pp 428–444, Sage, Thousand Oaks, California.

Unpublished reports/theses

Nandhakumar JJ (1993) *The practice of executive information systems development: and in-depth case study*. PhD Thesis, Department of Engineering, University of Cambridge.

PROOFS

Proofs of papers will be sent to authors for checking. Alterations to diagrams should be avoided where possible. It will not be possible to accept major textual changes at this stage. Proofs must be returned to the publishers within 48 hours of receipt by fax, first-class post, airmail or courier. Failure to return the proof will result in the paper being delayed.

PDF OFFPRINT

Corresponding authors will receive a PDF of their article. This PDF offprint is provided for personal use. It is the responsibility of the corresponding author to pass the PDF offprint onto co-authors (if relevant) and ensure that they are aware of the conditions pertaining to its use.

The PDF must not be placed on a publicly-available website for general viewing, or otherwise distributed without seeking our permission, as this would contravene our copyright policy and potentially damage the journal's circulation. Please visit http://www.apeiron-journals.com/JITA/authors/rights_and_permissions.html to see our latest copyright policy.

COPYRIGHT AND PERMISSIONS

The copyright of all material published in the Journal is held by Paneuropean University APEIRON. The author must complete and return the copyright form enclosed with the proofs.

Authors may submit papers which have been published elsewhere in a foreign language, provided permission has been obtained from the original publisher before submission.

Authors wishing to use material previously published in JITA should consult the publisher.

FINAL MATERIAL

All final material must be submitted electronically in its original application format (MS Word is preferred). The file must correspond exactly to the final version of the manuscript.

CORRESPONDENCE

Business correspondence and enquiries relating to advertising, subscriptions, back numbers or reprints should be addressed to the relevant person at:

Paneuropean University APEIRON
Journal JITA
Pere Krece 13, P.O.Box 51
78102 Banja Luka
Bosnia and Hercegovina / RS

PUBLICATION ETHICS

We take an active interest in issues and developments relating to publication ethics, such as plagiarism, falsification of data, fabrication of results and other areas of ethical misconduct. Please note that submitted manuscripts may be subject to checks using the corresponding service, in order to detect instances of overlapping and similar text.

JITA

PUBLISHER

Paneuropean University APEIRON,
College of Information Technology
Banja Luka, Republic of Srpska, B&H
www.apeiron-uni.eu

Darko Uremović, Person Responsible for the Publisher
Aleksandra Vidović, Editor of University Publications

EDITORS

Gordana Radić, PhD, Editor-in-Chief (B&H)
Zoran Ž. Avramović, PhD, (B&H)
Dušan Starčević, PhD, (B&H)

EDITORIAL BOARD

Zdenka Babić, PhD, (B&H)
Leonid Avramović Baranov, PhD, (Russia)
Patricio Bulić, PhD, (Slovenia)
Valery Timofeevič Domansky, PhD, (Ukraine)
Hristo Hristov, PhD, (Bulgaria)
Emil Jovanov, PhD, (USA)
Branko Latinović, PhD, (B&H)
Petar Marić, PhD, (B&H)
Vojislav Mišić, PhD, (Canada)
Boško Nikolić, PhD, (Serbia)
Dragica Radosav, PhD, (Serbia)
Gjorgji Jovanchevski, PhD, (Macedonia)
Mihaylovich Vladislav Yuryev, PhD, (Russia)

EDITORIAL COUNCIL

Siniša Aleksić, APEIRON University, Director
Risto Kozomara, APEIRON University, Rector

TECHNICAL STAFF

Lana Vukčević, Editorial Secretary
Stojanka Radić, Lector

EDITOR ASSISTENTS

Sretko Bojić, APEIRON University
Gordan Ružić, ETF University of Belgrade

ISSN 2232-9625



9 772232 962005