DOI: 10.5604/20830157.1130182

THE PROBLEM OF SYSTEM FAULT-TOLERANCE

Victor Mashkov[1], Andrzej Smolarz[2], Volodymyr Lytvynenko[3], Konrad Gromaszek[2]
[1]University J.E. Purkyne, Usti nad Labem, Czech Republic, [2]Lublin University of Technology, Institute of Electronics & Information Technology, Lublin, Poland,
[3]Kherson National Technical University, Department of Informatics & Computing Technology, Kherson, Ukraine

Abstract. *System level self-diagnosis (SLSD) has been deeply investigated in literature. It aims at diagnosing systems composed by units, which are required to be able to test each other by exchanging information through available links. The article describes a simplified state-transition diagram model which gives a general impression of how checking, diagnosis and recovery can "conjointly" influence the system reliability and fault-tolerance. The model uses the integrated parameters and is very useful as a starting point and is a basis for further refinements.*

Keywords: SLSD, system reliability

PROBLEM ODPORNOŚCI SYSTEMU NA USZKODZENIA

Streszczenie. *Autodiagnostyka na poziomie systemu jest szeroko opisywana w literaturze. Celem jest diagnostyka systemy składającego się z jednostek od których wymaga się aby miały możliwość wzajemnego testowania za pośrednictwem dostępnych połączeń. W artykule przedstawiono uproszczony model oparty na diagramie przejść który daje ogólny pogląd, jak sprawdzanie, diagnostyka i powrót do warunków normalnych mogą wspólnie wpływać na niezawodność systemu i odporności na uszkodzenia. Model wykorzystuje zintegrowane parametry i jest bardzo przydatne jako punkt wyjścia dla dalszych udoskonaleń.*

Słowa kluczowe: SLSD, niezawodność systemowa

Introduction

System level self-diagnosis (SLSD) was introduced by Preparata at al. [6] and has been deeply investigated in literature. It aims at diagnosing systems composed by units (optionally named processing elements), with the requirement that they are able to test each other by exchanging information through available links. At this level of diagnosis, each particular test is considered as atomic. It means that the details of a test are abstracted (not considered), and only the result of test is taken into consideration. Each test result is expressed via binary variable. It can take values either 0 or 1. The set of test results is called a *syndrome*. A syndrome contains information about the states of the system units in coded form. One of the tasks of SLSD is to decode a syndrome by using a diagnosis algorithm.

1. System level self-diagnosis

For providing system level self-diagnosis the tests among system units can be performed:

- either in accordance with a pre-set schedule (i.e., defined a priori).
- or in an adapted manner when, at the beginning, the tests are performed in accordance with defined a priori testing assignment. Once a unit is diagnosed as fault free, the tests it performs are considered reliable, and therefore, any other units should only be tested ones by this fault-free unit to correctly determine its status. Thus, the testing assignment is adapted such that units diagnosed as fault-free perform all the testing in the system [1].
- or entirely randomly (i.e., from the beginning to the end of testing).
- or adaptively randomly. At the beginning, all units are engaged in tests performing. Tests are performed randomly. Once a test result takes the value of 1, the units participated in this test (so-called suspected pair) should only be tested by other system units (i.e., should not perform tests on other units). The choice of each pair of units for testing is performed randomly.

In all cases, the intention is to minimize the time of performance of the set of tests. Random performing of tests is considered both in context of system self-checking and system self-diagnosis.

Self-checking is the process which aims at discriminating between two states of a system: fault-free and faulty. The result of self-checking doesn't indicate which of the system units has failed, and only testifies the presence of fault(s) in the system.

Self-checking may require small number of tests. When $P_{AT}=1$ and $P_S = P_F = 1$ (see Table 1), it is only needed to find out that each of the system units has been tested, at least, once. It may happen that N tests will be sufficient for system self-checking (see Fig. 1), where N is the number of system units.

Table 1. Test results and their probabilities

Test result and its probability		Testing unit u_i	
		fault-free	faulty
Tested unit u_j	fault-free	$r_{ij} = 0$ (P_C)	$r_{ij} = 0$ $(1 - P_S)$
		$r_{ij} = 1$ $(1 - P_C)$	$r_{ij} = 1$ (P_S)
	faulty	$r_{ij} = 0$ $(1 - P_{AT})$	$r_{ij} = 0$ $(1 - P_F)$
		$r_{ij} = 1$ (P_{AT})	$r_{ij} = 1$ (P_F)

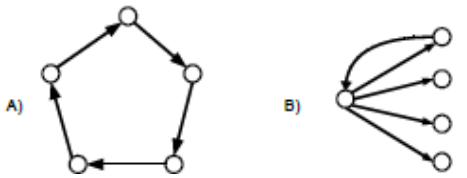


Fig. 1. Cases when each unit is tested

For providing system self-checking it is not necessary to form the syndrome at all cost, and, consequently, to perform its analysis. Only message or signal informing about system fault-free (resp. faulty) state is sufficient. This can be done, for example, by the unit which has produced the test result equal to 1. Further we are going to consider the case when tests are performed during the system operation. Hence, it is not possible to determine in advance which of the system units will be idle at the definite moment of time and, thus, will be able to test (or be tested by) another system unit. From this it follows that not only pair of units that provides a test, but also instance of test performing is random. The random value is also the number of tests which will be performed in the system during a certain period of time.

At the beginning, the self-checking procedure is performed to find out if the system possesses a faulty unit(s). The period of self-checking duration depends on the requirements to the credibility of self-checking result. If no test result equal to 1 is obtained during the self-checking (i.e., all test results are equal to 0), then the self-checking procedure ends, and the respective message or signal is delivered to the system environment. The self-checking procedure and subsequent delivering of information about the state of the system can be repeated at certain intervals as long as the system is operating. Otherwise. that is, when the test result indicating the presence

of a faulty unit in the system is obtained, the self-checking procedure is terminated immediately, and the procedure of self-diagnosis will be started. The aim of self-diagnosis procedure is to identify the faulty unit(s). As the research results show, one of the most difficult tasks is the task of determining the time duration of self-checking when all test results indicate that there are no faulty units in the system (i.e., all test results are equal to 0). In Fig. 2, the cycle of self-checking (SSC) and probable self-diagnosis are depicted.

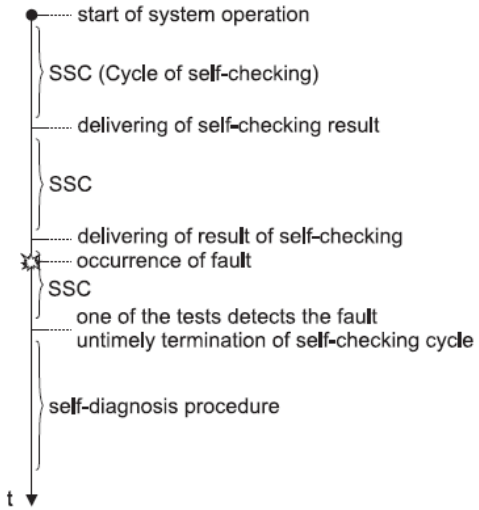


Fig. 2. Self-checking cycles and fault occurrence

Fig. 2 can also help to elucidate the important features of self-checking. From Fig. 2, it is seen that fault occurrence doesn't lead immediately to termination of self-checking procedure. Self-checking, as a rule, will continue until the fault is detected (captured) by one of the tests. After normal termination of each SSC, the result of self-checking is delivered to the system environment. This result indicates that the system is fault-free. Only in case of anomalous termination of SSC, no result of self-checking is delivered to the system environment. Thus, normally, the same information is delivered to the system environment. Consequently, the idea springs to mind, that self-checking could be organized in such way that its result will not be delivered at all. In this case, absence of information about system state would mean that system is fault-free. However, this proposition has not been enough researched both from the theoretical and practical points of view. Nevertheless, it is worth noting that this situation can be considered in context of our consideration as a particular case when the time duration of self-checking cycle approaches the infinite.

For organization of SSC (mainly, for defining the time duration of SSC) there were suggested several solutions [2, 3, 4]. Basically, SSC continues until one of the following conditions is met:

1) pre-set time has expired. Time duration of SSC is a constant value and is fixed in advance,
2) certain number of tests has been received. Time duration of SSC is defined by the certain number of performed tests, i.e., SSC continues until there is performed pre-set number of tests. Time duration of SSC is random,
3) certain diagnosis graph (DG) has been formed. SSC continues until the tests form a certain diagnosis graph (resp. DG which belongs to the subset of diagnosis graphs defined a priory). Time duration of SSC is random.

The cases when time duration of SSC is fixed or defined by a certain number of performed tests can be further described from the point of view of whether the analysis of the received diagnosis graph has to be performed or not. When such analysis doesn't have to be performed, the task arises to compute the probability of the event that all system units have been tested at least once. However, in practice there can be applied the opposite attitude when the time duration of SSC (resp., the required number of tests) is computed basing on the required probability of the event that all system units will be tested. Analysis of the obtained DG aims at checking whether all system units have been tested or whether the formed DG belongs to predefined subset of diagnosis graphs. It depends on the value of required credibility of self-checking result. When analysis shows that not all of the system units have been tested, it is possible to continue the SSC by the predefined period of time (so-called, extended period). After this extended period expires, the analysis is repeated. But this time, all of the tests both performed during the main and extended periods are accounted. Determining the optimal number of possible extended periods of SSC and the time of their duration is a separate problem.

2. System fault-tolerance

System tolerance to the failure of its units can be modeled by using different mathematic models. Mostly, for this purpose there is used the system state-transition diagram (Markov model).

Markov model is analyzed in order to determine the probability of system being in a given state at a given point in time, the amount of time a system is expected to spend in a given state, as well as the expected number of transitions between states. On the basis of these probabilities it is possible to quantify and estimate the system reliability and system fault-tolerance.

For the systems capable of graceful degradation the state-transition diagram includes the following states:

S_0 – all of the system units (i.e., N units) are actively engaged in performing system and diagnosis tasks. In other words, the system is fully operational,
S_1 – one of the system units is isolated (i.e., it doesn't perform system tasks). The system is minorly degraded, but still continues to deliver degraded (although acceptable) services,
S_2 – two system units are isolated. In the system, there remain N-2 active units. System is majorly degraded, but is still able to deliver acceptable services,
S_3 – total failure.

For simplicity reason, here only systems which can tolerate the presence of not more than two faulty units are considered. Transitions of a system from one state to another are depicted in Fig. 3.

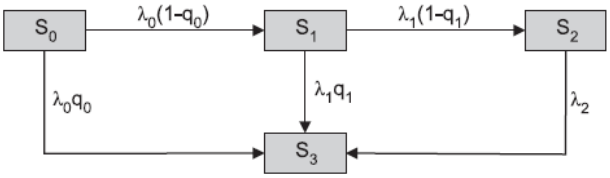


Fig. 3. Model of system fault-tolerance

By λ_0, λ_1, λ_2 are denoted rates of system transitions from one state to another, and by q_0, q_1 are denoted the probabilities of corresponding transitions. The values of λ_i , i = 0, 1, 2 depend on the reliability of system units, and the values of q_i , i = 0, 1 depend on the efficiency of self-checking, self-diagnosis and recovery procedures. Transitions between particular states can be considered following the Poisson model. Poisson model has proven suitable to describe many of natural and technical processes. Palm in [5] pointed out that in many cases the superposition of a large number of independent stationary processes can be approximated by a Poison process. This gives us the reason to apply the Poison model to system state-transition diagram under consideration. Since in Poisson model the waiting time (until the next occurrence of the event) follows an exponential distribution, the period of time of system being in a given state also has exponential distribution.

Let $P_i(t)$ be the probability of system being in state S_i at point in time t. Then, $\sum_{i=0}^{3} P_i(t) = 1$.

When transitions from one state to another follow the Poisson model the sought probabilities P_i, i = 0÷3, can be determined from the Kolmogorov equations:

$$\frac{\partial P_0(t)}{\partial t} = -\lambda_0 P_0(t)$$
$$\frac{\partial P_1(t)}{\partial t} = -\lambda_1 P_1(t) + \lambda_0(1-q_0)P_0(t)$$
$$\frac{\partial P_2(t)}{\partial t} = -\lambda_2 P_2(t) + \lambda_1(1-q_1)P_1(t)$$
$$\frac{\partial P_3(t)}{\partial t} = \lambda_0 q_0 P_0(t) + \lambda_1 q_1 P_1(t) + \lambda_2 P_2(t)$$

Kolmogorov equations describe the dynamics of entering the particular state, resp. leaving the particular state. For example, for state S_1 this dynamics is expressed by differential equation

$$\frac{\partial P_1(t)}{\partial t} = -\lambda_1 P_1(t) + \lambda_0(1-q_0)P_0(t)$$

It means that the system is leaving (sign minus) the state S_1 with intensity λ_1 and entering the state S_1 (sign plus) with intensity $\lambda_0(1-q_0)$. The state S_0 is the initial state. That is, $P_0(t=0)=1$, and $P_i(t=0)=0$ for i = 1,2,3. Taking Laplace transforms of Kolmogorov equations yields the following system of equations

$$sP_0(s) - 1 = -\lambda_0 P_0(s)$$
$$sP_1(s) = -\lambda_1 P_1(s) + \lambda_0(1-q_0)P_0(s)$$
$$sP_2(s) = -\lambda_2 P_2(s) + \lambda_1(1-q_1)P_1(s)$$
$$sP_3(s) = \lambda_0 q_0 P_0(s) + \lambda_1 q_1 P_1(s) + \lambda_2 P_2(s)$$

After solving this system of equations for $P_i(s)$, i=0, 1, 2, we receive

$$P_0(s) = \frac{1}{s+\lambda_0}$$
$$P_1(s) = \frac{\lambda_0(1-q_0)}{(s+\lambda_0)(s+\lambda_1)}$$
$$P_2(s) = \frac{\lambda_0\lambda_1(1-q_0)(1-q_1)}{(s+\lambda_0)(s+\lambda_1)(s+\lambda_2)}$$

It can be easily noticed that single equations can be expressed as

$$P_i(s) = \frac{\prod_{j=0}^{i-1}(1-q_j)\lambda_j}{\prod_{j=0}^{i}(s+\lambda_j)}$$

For inverse Laplace transform the following expression can be used

$$z(s) = \frac{1}{\prod_{i=1}^{m}(s+k_i)} \quad (Laplace\,s-domain) \rightarrow$$
$$z(t) = \sum_{\substack{i=1 \\ i\neq n}}^{m} \frac{e^{-k_i t}}{\prod_{\substack{n=0 \\ n\neq i}}^{m}(k_n - k_i)}, \quad m \geq 1 \quad (time\,domain)$$

Taking inverse Laplace transforms, these become

$$P_0(t) = e^{-\lambda_0 t}$$
$$P_1(t) = \frac{\lambda_0(1-q_0)}{\lambda_0-\lambda_1}(e^{-\lambda_1 t} - e^{-\lambda_0 t})$$
$$P_2(t) = \frac{\lambda_0\lambda_1(1-q_0)(1-q_1)}{(\lambda_0-\lambda_1)(\lambda_1-\lambda_2)(\lambda_0-\lambda_2)} \cdot$$
$$\cdot(\lambda_0 e^{-\lambda_2 t} - \lambda_1 e^{-\lambda_2 t} - \lambda_0 e^{-\lambda_1 t} + \lambda_2 e^{-\lambda_1 t} + \lambda_1 e^{-\lambda_0 t} - \lambda_2 e^{-\lambda_0 t})$$
$$P_3(t) = 1 - \sum_{i=0}^{2} P_i(t)$$

The probabilities of the system being in states S_0, S_1, S_2 and S_3, i.e., $P_0(t)$, $P_1(t)$, $P_2(t)$ and $P_3(t)$ are functions of time and some other parameters (λ and q). In its turn, probabilities q_0 and q_1 depend considerably on the efficiency of the checking, diagnosis and recovery procedures. Fig. 4 shows the impact of values of q_0 and q_1 on the probability $P_3(t)$.

Function $P_3(t)$ was calculated for the homogeneous system with five units which have $\lambda=10^{-4}$ 1/h. The case of $q_0=q_1=0$ corresponds to "absolutely perfect" checking, diagnosis and recovery. This probability $P_3(t)$ allows also to estimate the amount of time the system is expected to spend in states other than S_3 (i.e., time to failure). Mostly, the time while system is operating without maintenance is relatively short (relative to its mean time to failure). Hence, the impact of checking, diagnosis and recovery on the reliability of system is essential. For the systems with a great number of units it is difficult to provide detailed examination of their state-transition diagrams for determining all the above mentioned probabilities. Usually, only the main reliability and fault-tolerance parameters are determined. The most common reliability parameter is the mean time to failure (MTTF), which can also be specified as the failure rate or the number of failures during a given period. The MTTF is usually specified in hours, but can also be used with other units of measurement (e.g., in cycles).

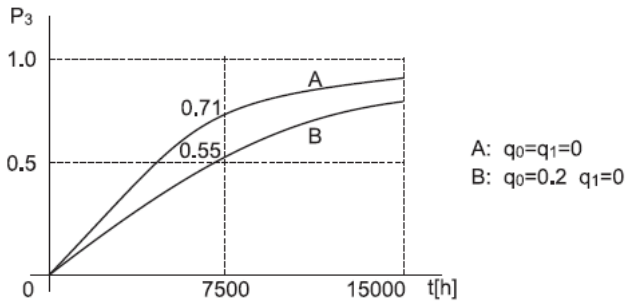


Fig. 4. Function $P_3(t)$

MTTF, T_0, can be calculated by using Tauberian theorem according to which

$$T_0 = \lim_{s\to 0} P_\Sigma(s), \text{ where } P_\Sigma(s) = \sum_{i=0}^{2} P_i(s)$$

For the system under consideration T_0 is equal to

$$T_0 = \sum_{i=0}^{2} P_i(s)\Big|_{s=0} =$$
$$= \frac{1}{\lambda_0} + \sum_{i=1}^{2} \frac{\prod_{k=0}^{i-1}(1-q_k)}{\lambda_i} = \frac{1}{\lambda_0} + \frac{(1-q_0)}{\lambda_1} + \frac{(1-q_0)(1-q_1)}{\lambda_2}$$

In Fig. 5, the dependence of T_0 on q_0 and q_1 is shown for the system with N=5 and $\lambda=10^{-4}$ 1/h.

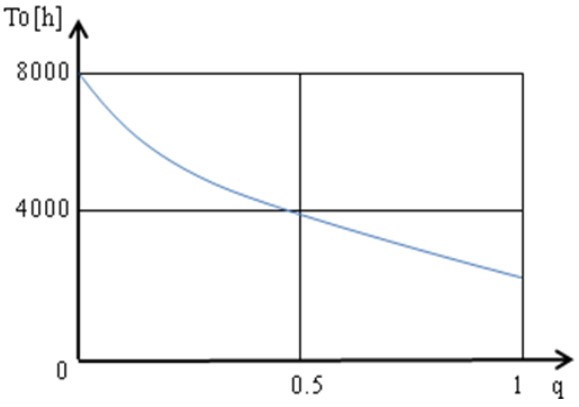


Fig. 5. Dependence of T_0 on $q_0 = q_1 = q$.

From Fig. 5 it can be seen to what extent the improving the checking, diagnosis and recovery can influence the value of MTTF. The next system reliability parameter is the probability of fault-free operation during the time t. This probability, $P_B(t)$,

can be expressed as the sum of probabilities of system being in all states except the state of total failure. That is,

$$P_B(t) = \sum_{i=0}^{2} P_i(t)$$

For a system which is unable to tolerate the failures of single units, the event of system leaving the state S_0 leads immediately to system failure (i.e., direct transmission into state S_3). From this we can deduce that the period of time when the system is being in states S_1 and S_2 reflects the system ability to tolerate the failures of its units. The mean time of this period, T_ω, can be calculated as follows

$$T_\omega = \lim_{s\to 0} P_\omega(s), \text{ where } P_\omega(s) = \sum_{i=1}^{2} P_i(s)$$

As an indicator of system fault-tolerance, it is normally used the total number of failed units which system can tolerate and continue in delivering acceptable services. As another indicator of system fault-tolerance, there can be used the following ratio:

$$Q = \frac{T_\omega}{T_0}$$

For the model under consideration the indicator Q is equal to

$$Q = \frac{R}{\frac{1}{\lambda_0} + R}, \text{ where } R = \sum_{i=1}^{2} \frac{\prod_{k=0}^{i-1}(1-q_k)}{\lambda_i}$$

Dependence of Q on q_0=q_1=q is depicted in Fig. 6.

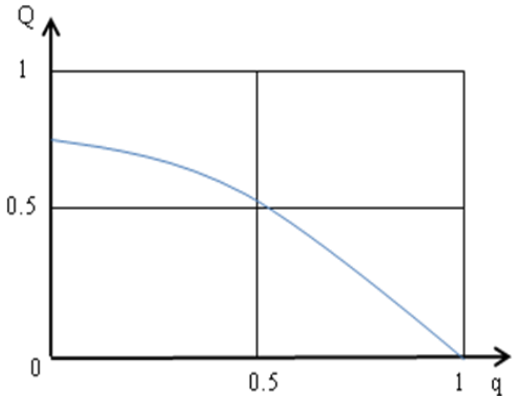


Fig. 6. Dependence of Q on q

In order to elucidate how the indicator Q characterizes the system fault-tolerance, let us consider two systems. Assume that both systems have the equal value of MTTF, i.e., T_0^1=T_0^2. Assume also that the first system has Q=0.2 and the second one has Q=0.8. In this case, we can conclude that the first system has reliable units but not very effective means of checking, diagnosis and recovery. In contrast, the second system has not very reliable units but has very effective means of checking, diagnosis and recovery. In case of $T_0^1 \neq T_0^2$, the system fault-tolerance can be evaluated by value of T_ω. However, in this case we can make only rough estimate.

3. Conclusions

It should be noted, that the above considered model (state-transition diagram) is very much simplified and only gives general impression of how checking, diagnosis and recovery can "conjointly" influence the system reliability and fault-tolerance. The model uses the integrated parameters (e.g., probabilities q_i). It means that, by using this model, it is difficult to decide on what specific measures should be undertaken in order to increase these probabilities to a certain value. This model doesn't allow to estimate to what extent increasing the efficiency of each procedure (checking, diagnosis, recovery) improves the system reliability and fault-tolerance. Nevertheless, this simplified model is very useful as a starting point and is a basis for further refinements.

References

[1] Bianchini, R.; Buskens, R.: An adaptive distributed system-level diagnosis algorithm and its implementation. In the 21st International IEEE Symposium on Fault-Tolerant Computing. New York, 1991, 222-229.

[2] Gostev, V.; Mashkov, V.; Mashkov, O.: Self-diagnosis of modular systems in random performance of elementary tests. Cybernetics and Computing Technology (Discrete Control System). No.105, 1997, 104-111.

[3] Mashkov, V.: Identification of air-borne control computer systems technical state on the basis of cross-testing. Proc. 5th Mini Conf. on Vehicle System Dynamics. Budapest, 1996, 78-88.

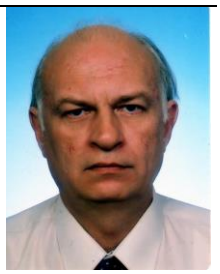
[4] Mashkov, V.; Barabash, O.: Self-checking of modular systems under random performance of elementary checks. Engineering Simulation. Vol.12, 1995, 433-445.

[5] Palm, C.: Intersitatsschwankungen in Fernspreshverkehr. Ericsson Technics. 44, 1943.

[6] Preparata, T.; Metze, G.; Chien, R.: On the connection assignment problem of diagnosable system. IEEE Transactions on Electronic Computers. Vol. EC-16, No.12, 1967, 848-854.

Prof. Victor Mashkov
e-mail: viktor.mashkov@ujep.cz

Doctor of Sciens in Engineering Docent Department of IT at the University of J.E. Purkyne in Usti nad Labem (Czech Republic). His major research focuses on dependability of computer systems, software fault tolerance, system level self-diagnosis and multi-agent systems



Ph.D. Andrzej Smolarz
e-mail: a.smolarz@pollub.pl

Assistant professor in Institute of Electronics and Information Technologies at Lublin University of Technology. Research field covers wide variety of optical methods in industrial diagnostics and control as well as applications of artificial intelligence methods in industrial diagnostics. Author of nearly 100 publications in this research area.

Prof. Volodymyr Lytvynenko
e-mail: immun56@gmail.com

Kherson National Technical University. D.Sc. (Eng. hab.), Professor, Head of the Department Informatics & Computer Sciences. Research interests: artificial immune systems, time series forecasting, multifractal analysis, reinforcement learning..

Ph.D. Konrad Gromaszek
e-mail: k.gromaszek@pollub.pl

Assistant professor in Institute of Electronics and Information Technologies at Lublin University of Technology. IEEE member. Research interests: hierarchical and adaptive control algorithms, computer networks, ICT systems, digital signal processing, PACs, DAQs, databases and data mining.

otrzymano/received: 2014.10.06 *przyjęto do druku/accepted: 2014.11.14*