19

The Decoding Algorithms as Techniques for Creation the Anomaly Based Intrusion Detection Systems

Evgeniya Nikolova and Veselina Jecheva
Burgas Free University, Faculty for Computer Science and Engineering
Bulgaria

1. Introduction

Since information and related infrastructure have become a very important part of today's companies and public organizations' networks, increasing attention has been given to security policies and mechanisms, that can help minimize the risk of unauthorized access and availability threats. Intrusion Detection Systems (IDS) are among the most disseminated security tools usually applied in order to detect attacks. These systems are categorized into misuse detection and anomaly detection systems (Ghosh et al., 1999). Misuse IDSs detect known attacks using preliminarily defined intrusion patterns and signatures in the system activity data. This method is similar to the approach most antivirus programs detect malware.

Anomaly-based approaches in IDS have the advantage of being able to detect unknown attacks since they look for patterns that deviate from the normal behavior (Bahrololum&Khaleghi, 2008). These systems lie on the assumption that an intrusion can be detected by observing a deviation from the normal or expected behavior of the system or network. They monitor network traffic and compare it against a preliminarily established baseline. The baseline describes what behavior is considered to be "normal" for that system and any activity, which deviates significantly from this baseline, is considered to be anomalous. The anomaly IDS have the following advantages over misuse detection approaches: can detect attempts to exploit new and unforeseen vulnerabilities without specific knowledge of details; can detect 'abuse-of-privilege' types of attacks, which usually do not exploit any security vulnerabilities and can recognize unusual network traffic based on network packet characteristics. The major challenges that anomaly IDS have to solve are the improvement of the detection process and the reduction of the number of the false alarms (Dagorn, 2008).

2. Outline of the methodology

The task of an intrusion detection system (IDS) is modelled as a classification problem in a machine-learning context. A typical anomaly recognition model will analyze data, compare to a known profile, run statistical analysis to determine if any deviation is significant, and flag the event(s) as a normal activity or an attack. This problem is very similar to the problem of decoding in the coding theory, that's why we consider this recognition as a decoding problem and we apply the well-known techniques as the Bahl-Cocke-Jelinek-Raviv (BCJR or the MAP) decoding algorithm or the max log MAP algorithm and the junction tree algorithm (JTA). First, we described the system using an oriented graph with nodes – the system state and edges - the system states transitions and applied BCJR algorithm as a method for intrusion detection during the system work. The second present method consists of two stages – the first contains the HMM creation and its adjustment using the gradient method, and the second one includes the intrusion recognition using the decoding algorithm – BCJR or the max-log-MAP algorithm. The forwards-backwards algorithm, also known as the BCJR, for HMM is equivalent to the JTA. The third presented method applies the JTA for the intrusion detection. More details about the results, which are obtained by the methodology based on the enumerated algorithms, are presented in our previous works (Jecheva& Nikolova, 2007, Nikolova&Jecheva, 2007, Nikolova& Jecheva, 2008).

2.1 The system model

As it was already outlined in the introduction, anomaly IDS models operate by building a model of "normal" system behavior. Normal system behavior is determined by observing the standard activity of the system, which has to be protected. In anomaly intrusion detection, how to model the normal behavior of activities performed by a user is an important issue. To extract the normal behavior as a profile, conventional data mining techniques are widely applied to a finite audit data set.

There are various methods for describing the legal user activities. One of them is the Hidden Markov Model (HMM) (Rabiner 1989, Qiao et al., 2002, Vigna 2003, Joshi 2005; Tan 2008), which provides a unifying framework for many tasks, where a measure of uncertainty is needed. The formal definition of a HMM is as follows: $\lambda = (A, B, \pi)$, where A is the state transition probability matrix, B is the observation probability distribution and the vector $\pi = (\pi_1, \pi_2, ..., \pi_N)$ is the initial probability distribution.

Let $S=(S_1, S_2, \ldots, S_N)$ be our state alphabet set, and $V= (v_1, v_2, \cdot\cdot\cdot, v_M)$ is the observation alphabet set. We define $Q=(q_1, q_2, \ldots, q_T)$ to be a fixed state sequence of length T, and corresponding observations $O=(O_1, O_2, \ldots O_T)$, where each O_t is a certain element $v_k \in V$. The square matrix $A=\{a_{ij}, 1 \le i \le N, 1 \le j \le N\}$, $0 \le a_{ij} \le 1$ and $\sum_{j=1}^{N} a_{ij} = 1$ contains elements, which represent the probability of transitioning from a given state to another possible state. The observation probability distribution is a non-square matrix $B=\{b_j(O_k), 1 \le j \le N, 1 \le k \le M\}$, with dimensions number of states by number of observations. It represents the probability that a given observable symbol will be emitted by a given state.

We consider those processes only in which the state transition probabilities do not change with time, i.e. $P(q_t = S_j \mid q_{t-1} = S_i) = a_{ij}$ the probability of transiting from state S_i to state S_j does not depend on the moment of time t (stationarity assumption) and depends on the previous state only (first-order HMM).

The main goal of the HMM is to describe the system behavior during specific period of time. In order to achieve this goal we determine the model parameters A, B and π for given

HMM λ such that $L = P(O|\lambda)$ takes the maximal value for the observation sequence *O*. This problem is known as learning problem. There are several optimization criteria for learning, out of which a suitable one is selected depending on the application. We apply the Maximum Likelihood (*ML*) as optimization criteria.

2.2 ML criterion

The ML criterion is based on the gradient based method, in which any parameter Θ of the HMM λ is updated according to the standard formula

$$\Theta_{new} = \Theta_{old} - \eta \left[\frac{\partial J}{\partial \Theta} \right]_{\Theta=\Theta_{old}}, \tag{1}$$

where *J* is a quantity to be minimized. In our case we set $J = -\log p\left(O|\lambda\right) = -\log L$. The minimization of *J* is equivalent to the maximization of *L*. We have

$$L = \sum_{i=1}^{N} p\left(O, q_t = i|\lambda\right) = \sum_{i=1}^{N} \alpha_t\left(i\right)\beta_t\left(i\right), \tag{2}$$

where the forward variable $\alpha_t\left(i\right)$ can be calculated using the following recursive steps:

$$\begin{gathered} \alpha_1(j) = \pi_j b_j(O_1),\ 1 \le j \le N, \\ \alpha_{t+1}(j) = p(O_1, O_2, ..., O_t, q_t = S_i \mid \lambda) = b_j(O_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij},\ 1 \le j \le N,\ 1 \le t \le T-1 \end{gathered} \tag{3}$$

and the backward variable $\beta_t\left(i\right)$ can be calculated efficiently recursively as follows:

$$\begin{gathered} \beta_T(i) = 1,\ 1 \le i \le N, \\ \beta_t(i) = p(O_{t+1}, O_{t+2}, ..., O_T, q_t = S_i \mid \lambda) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}),\ 1 \le i \le N,\ 1 \le t \le T-1 \end{gathered} \tag{4}$$

Since there are two main parameter sets in the HMM, transition probabilities a_{ij} and observation probabilities $b_j(O_k)$, we can find the gradient $\frac{\partial J}{\partial \Theta}$ for each of the parameter sets.

- Gradient with respect to the transition probabilities

$$\frac{\partial J}{\partial a_{ij}} = -\frac{1}{L} \sum_{t=1}^{T} \beta_t\left(j\right) b_j\left(O_t\right) \alpha_{t-1}\left(i\right). \tag{5}$$

- Gradient with respect to the observation probabilities

$$\frac{\partial J}{\partial a_{ij}} = -\frac{1}{L}\frac{\alpha_t(j)\beta_t(j)}{b_j(O_t)}. \tag{6}$$

2.3 The BCJR algorithm

The BCJR decoding algorithm estimates random parameters with prior distributions. In the case examined the algorithm scans the traces of the system activity and compares the current activity with the patterns of normal user activity (Bahl et al., 1974). If the deviation from the normal data of system activity is above the preliminarily defined threshold, then the current system call is marked as abnormal, i.e. an intrusion is detected. The description of the BCJR algorithm can be performed based on log-likelihood ratios (*LLR*). The *LLR* are represented as follows:

$$LLR = \ln\frac{P\left(m_i = 1 \middle| O_i\right)}{P\left(m_i = 0 \middle| O_i\right)} \tag{7}$$

where m_i is the message bit associated with the state transition q_i to q_{i+1} and $P\left(m_i = 1 \middle| O_i\right)$ is the a posteriory probability in which the bit, determining the presence of attack, is equal to 1. If the *LLR* of an observation is positive, it implies that m_i is most likely to be a 1 and if it is negative, m_i is most likely to be zero. The algorithm consists of three steps:

- Forward recursion. The forward state metrics $\alpha_t(s_t)$ represent the probability that the current state is s_t given the noisy observation vector $(O_1,...,O_t)$ and are recursively calculated

$$\alpha_0(s_0) = \begin{cases} 1, & s_0 = 0 \\ 0, & otherwise \end{cases},$$
$$\alpha_t(s_t) = \sum_{s_{t-1}, i=0,1} \alpha_{t-1}(s_{t-1})\gamma_i(O_t, s_{t-1}, s_t). \tag{8}$$

- Backward recursion. The backward state metrics $\beta_t(s_t)$ represent the probability that the final state is s_{t+1} and are recursively calculated

$$\beta_M(s_M) = \begin{cases} 1, & s_M = 0 \\ 0, & otherwise \end{cases},$$
$$\beta_t(s_t) = \sum_{s_{t+1}, i=0,1} \gamma_i(O_{t+1}, s_t, s_{t+1})\beta_{t+1}(s_{t+1}). \tag{9}$$

- Log-Likelihood Ratios

$$LLR\left(m_t\right)=\ln\frac{\sum\limits_{s_t,s_{t-1}}\alpha_{t-1}\left(s_{t-1}\right)\gamma_0\left(O_t,s_{t-1},s_t\right)\beta_t\left(s_t\right)}{\sum\limits_{s_t,s_{t-1}}\alpha_{t-1}\left(s_{t-1}\right)\gamma_1\left(O_t,s_{t-1},s_t\right)\beta_t\left(s_t\right)},\tag{10}$$

$$\gamma_i\left(O_t,s_{t-1},s_t\right)=q\left(m_t=i\left|s_t,s_{t+1}\right.\right)P\left(O_t\left|m_t=i,s_{t-1},s_t\right.\right)P\left(s_t\left|s_{t-1}\right.\right).\tag{11}$$

2.4 Max log MAP algorithm

The $\alpha_t\left(s_t\right)$ and $\beta_t\left(s_t\right)$ parameters in the MAP algorithm are approximated in the max-log-MAP algorithm by maximization operation (Robertson et al., 1995, Benedetto et al., 1997). The estimated *LLRs* are computed by exhaustively exploring all possible state transitions from s_{t-1} to s_t using forward and backward recursion.

- Forward recursion. The forward state metrics $\alpha_t\left(s_t\right)$ are recursively calculated

$$\begin{gathered}\alpha_0\left(s_0\right)=\begin{cases}1, & s_0=0\\0, & otherwise\end{cases},\\ \alpha_t\left(s_t\right)=\max_{s_{t-1},i=0,1}\alpha_{t-1}\left(s_{t-1}\right)\gamma_i\left(O_t,s_{t-1},s_t\right).\end{gathered}\tag{12}$$

- Backward recursion. The backward state metrics $\beta_t\left(s_t\right)$ are recursively calculated

$$\begin{gathered}\beta_M\left(s_M\right)=\begin{cases}1, & s_M=0\\0, & otherwise\end{cases},\\ \beta_t\left(s_t\right)=\max_{s_{t+1},i=0,1}\gamma_i\left(O_{t+1},s_t,s_{t+1}\right)\beta_{t+1}\left(s_{t+1}\right).\end{gathered}\tag{13}$$

- *LLR* computation. The output for each bit at time t is computed by using the backward state metrics $\beta_t\left(s_t\right)$ and the corresponding forward state metrics $\alpha_t\left(s_t\right)$ as follows

$$LLR\left(m_t\right)=\ln\frac{\max\limits_{s_t,s_{t-1}}\alpha_{t-1}\left(s_{t-1}\right)\gamma_0\left(O_t,s_{t-1},s_t\right)\beta_t\left(s_t\right)}{\max\limits_{s_t,s_{t-1}}\alpha_{t-1}\left(s_{t-1}\right)\gamma_1\left(O_t,s_{t-1},s_t\right)\beta_t\left(s_t\right)}.\tag{14}$$

2.5 The junction tree algorithm

The JTA is an inference algorithm for any graphical model, which gives a solution for the following problem: calculating the conditional probability of a node or a set of nodes, given the observed values of another set of nodes. The idea of this algorithm is to find ways to decompose a global calculation on a joint probability into a linked set of local computations. The algorithm consists of the following steps (Lauritzen at al., 1988, Lauritzen , 1996):

- Given directed graph is converted into an undirected graph G, so an uniform treatment of directed and undirected graphs is possible.
- Form a triangulated graph $\widetilde{G}$ by adding edges as necessary. (Chord is a link joining two non-consecutive vertices of a loop. An undirected graph is triangulated if every loop of length 4 or more has a chord.)
- Given a triangulated graph, a junction tree is constructed by forming a maximal spanning tree from the cliques in that graph. A clique is a subset of vertices containing only one vertex or such that any two vertices are neighbours. A clique tree (in which nodes are cliques of the triangulated graph) will be constructed with separators.
- Extract a junction tree. A clique tree is a junction tree if it has the following two properties:
    - singly connected: there is exactly one path between each pair of nodes;
    - running intersection: all nodes on the path between v and w contain the intersection $v \cap w$.
- Run sum-product, which is the basic decoding algorithm for nodes on graphs, on the resulting junction tree.

Except finding marginal probabilities, JTA helps to answer another natural question: what is the most likely state of the distribution? In our case the considered junction tree is presented in Figure 1:

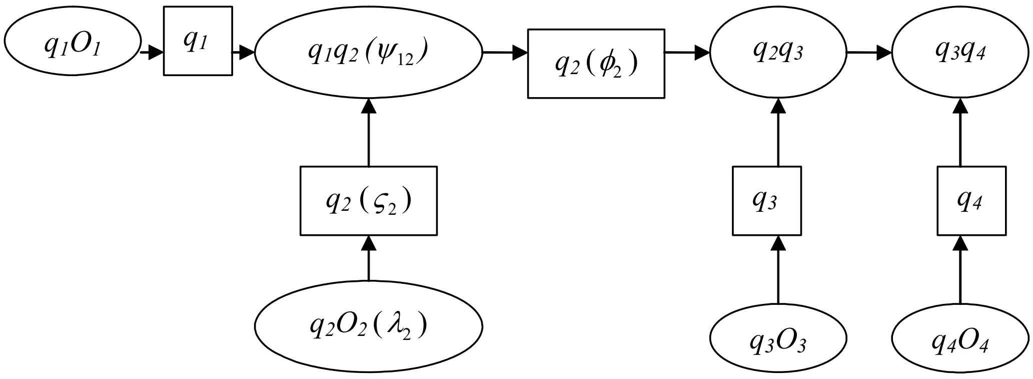

Fig. 1. The junction tree representing the system work

where the vector $O=(O_1, O_2, \ldots, O_T)$ is the current observation sequence and $Q=(q_1, q_2, \ldots, q_T)$ is the state sequence at the moments $t=1, 2, \ldots, T$. Each q_t is one of the elements of the set $S = \{s_1, \ldots, s_N\}$.

As a first step, all clique and separator potentials have been initialized with 1^s. Then the conditional probability of each node in the original graph is multiplied onto the clique to which it is assigned. If we assume initially that the nodes, containing observations O_t, are hidden, the λ potentials at the node q_tO_t will be all 1^s and the potentials along the node q_tq_{t+1} will be as follows:

$$\Psi_{1,2}(i,j) = P\left(q_2 = s_j \left| q_1 = s_i \right.\right) P\left(q_1 = s_i\right) \tag{15}$$

$$\Psi_{t,t+1}(i,j) = P\left(q_{t+1} = s_j \left| q_t = s_i \right.\right) \tag{16}$$

- Forward steps. The potentials were initialized with evidence

$$\lambda_t^*(i) = P\left(O_t \left| q_t = s_i \right.\right) = \varsigma_t^*(i) \tag{17}$$

Running the forward algorithm, the following potentials were obtained:

$$\psi_{t-1,t}^*(i,j) = P(q_{t-1}, q_t, O_{1:t}) = \psi_{t-1,t}(i,j)\phi_{t-1}^*(i)\varsigma_t^*(j). \tag{18}$$

Marginalizing yields:

$$\phi_t^*(j) = P\left(q_t = s_i, O_{1:t}\right) = \sum_i \psi_{t-1,t}^*(i,j). \tag{19}$$

- Backward steps. In the backward steps, the potentials are

$$\psi_{t-1,t}^{**}(i,j) = P\left(q_{t-1} = s_i, q_t = s_j, O_{1:T}\right) = \frac{\psi_{t-1,t}^*(i,j)}{\phi_t^*(j)}\phi_t^{**}(j). \tag{20}$$

Marginalizing yields:

$$\phi_t^{**}(j) = P\left(q_t = s_j, O_{1:T}\right) = \sum_i \psi_{t-1,t}^{**}(i,j). \tag{21}$$

If the messages were propagated from children to their parents in the rooted junction tree, the result of the induction after the forwards steps (after collecting to root) will be $\psi_{t,t+1}^*(i,j) = P\left(q_t = s_i, q_{t+1} = s_j\right)$, $\phi_t^*(i) = P\left(q_t = s_i\right)$ and all the other potentials will be unchanged (all 1s). After the backward steps we have $\psi_{t-1,t}^{**}(i,j) = P\left(q_{t-1} = s_i, q_t = s_j\right)$, $\varsigma_t^{**}(i) = \lambda_t^{**}(i) = \phi_t^{**}(i) = P\left(q_t = s_i\right)$. The result from the JTA is the most likely state sequence during the examined period of time, whereupon the state transition probabilities can easily be computed from the result state sequence.

3. Simulation experiments and statistical methods of evaluating the effectiveness of IDS

3.1 Simulation experiments

A large number of simulation experiments, which were based on the described model, were carried out in order to test the proposed methodology. The experimental data were obtained from Computer Immune Systems Project (University of New Mexico), performed by the researches in the Computer Science Department, University of New Mexico.

The simulation data are collected from Unix system examination during a period of time and consist of system call sequences, which were obtained from observation of some privileged processes executed on behalf of the root account as well some anomalous data. Each data file contains sequences of system call numbers, obtained by the examined process activity. The input data files are sequences of ordered pairs of numbers, where each line consists of one pair. The first number in each pair is the process ID (PID) of the process executed, and the second one is the system call number. Forks are taken into account as separate processes and their execution results are considered as normal user activity.

The privileged processes are among the major targets of the attacker as they are granted access to system resources that are inaccessible to ordinary users. The methods for pattern generation are described in (Forrest at al., 1996, Forrest at al., 1998). They prove the short sequences of system calls can be successfully applied for discriminating between normal and anomalous activities in the system.

The normal activity data patterns compose the system states set *S* and the intrusion activity patterns compose the set *V*. The system model, which contains the transition probabilities, was created according to the normal and anomalous data sets. This model is considered as a database, describing normal system activity. Each of the decoding algorithms, which were described in section 2, was applied during the detection stage in order to distinguish normal traces from abnormal ones.

3.2 The results obtained by the BCJR algorithm

The experimental data include normal user activity traces as well as intrusion data. A slide window with length *T* was applied in order to cross the traces of current user activity, i.e. the system observations, which compose the set *O*. The experiments were accomplished with the values of *T*=10, 15 and 20. The *LLR*s which represent the probability of intrusion occurrence at the given moment of time were obtained as a result of BCJR algorithm. Each *LLR* is the logarithmic ratio of the probability of attack presence and the probability of normal activity at specific moment *t*. Figure 2 presents some of the results of the *LLR*s for the process synthetic sendmail for *T*=20. The positive values of *LLR* denote an attack presence, while the negative values imply that the examined system call is a pattern of normal activity.

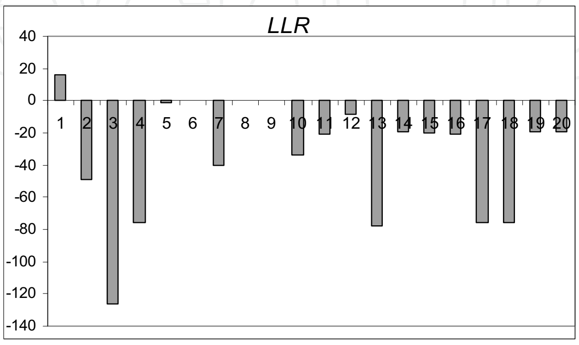

Fig. 2. Some results for *LLR* for the process synthetic sendmail for T=20

Another strategy for checking whether the particular activity data is normal or anomalous is prior to determine the model parameters *A*, *B* for given HMM λ and given sequence of observations *O* using standard gradient method and then to apply the BCJR decoding algorithm. Given an unknown observation sequence, the *ML*-criterion finds the model λ which maximizes the value of $L = P(O|\lambda)$. For standard gradient descent learning rate η were used the following values: from 0,000001 to 0,000009 with step 0,000001, from 0,00001 to 0,00009 with step 0,00001 and from 0,0001 to 0,0009 with step 0,0001 for both observation and transition probabilities. Some of the results for the process synthetic sendmail are summarized in Table 1, which present the number of iterations and the values of *L* for η=0,0001-0,0005 and η=0,00001-0,00005 and *T*=10 or 15.

T=10			*T*=15		
η	Number of iterations	*L*	η	Number of iterations	*L*
η=0,0001	322	4.20353e-13	η=0,0001	322	2.88298e-17
η=0,0002	161	3.41980e-13	η=0,0002	162	1.88268e-17
η=0,0003	107	3.21028e-13	η=0,0003	108	4.77443e-18
η=0,0004	80	7.60207e-14	η=0,0004	82	8.53916e-18
η=0,0005	64	1.15635e-13	η=0,0005	65	5.47520e-19
η=0,00001	3224	3.94035e-13	η=0,00001	3207	2.54814e-17
η=0,00002	1611	3.78713e-13	η=0,00002	1604	2.47694e-17
η=0,00003	1074	3.87351e-13	η=0,00003	1070	2.46345e-17
η=0,00004	805	3.58734e-13	η=0,00004	802	2.22364e-17
η=0,00005	644	3.58219e-13	η=0,00005	642	2.21205e-17

Table 1. Numbers of iterations and the values of *L* depending on the values of η

The algorithm exhibits a tendency to growth of the number of iterations when we increase the number of observations and decrease the learning rate η. The number of iterations necessary for the model training is similar when *T*=10 and 15. One of the greatest problems in training large models with gradient descent is to find an optimal learning rate. A small one will slow down the speed and significantly increase the number of iterations. On the other hand, a large one will probably cause oscillations during training and finally leading to no useful model would be trained.

Anomalous data was examined using the BCJR decoding algorithm which compares the traces of the system activity for *T*=10, 15 and 20 with the patterns of normal user activity. The intrusion detection problem is considered as a decoding problem. The results of the algorithm are the values of *LLR*, where each *LLR* is the logarithmic ratio of the probability of attack presence and the probability of normal activity at specific moment *t*. We assume that the values of *LLR* greater than 0 denote an attack presence. Some of the results for *T*=10, η=0,00001-0,00009 and the input files: synthetic ftp, named and xlock are summarized in the Figures 3, 4 and 5. For instance, from Figure 3 one can see that the method registers O_2 and O_4, as possible attacks.

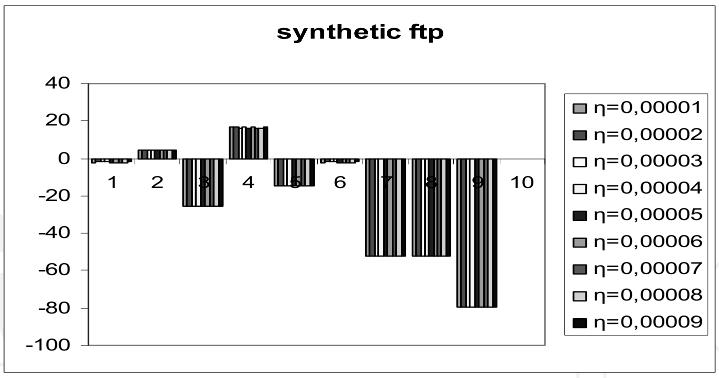

Fig. 3. Values of LLR depending on the value of η when T=10 for synthetic ftp

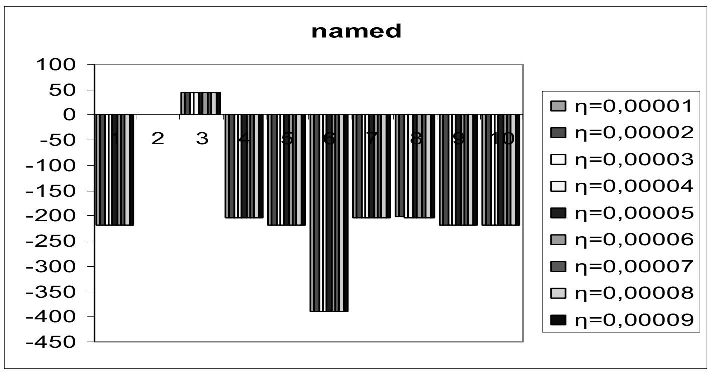

Fig. 4. Values of LLR depending on the value of η when T=10 for named

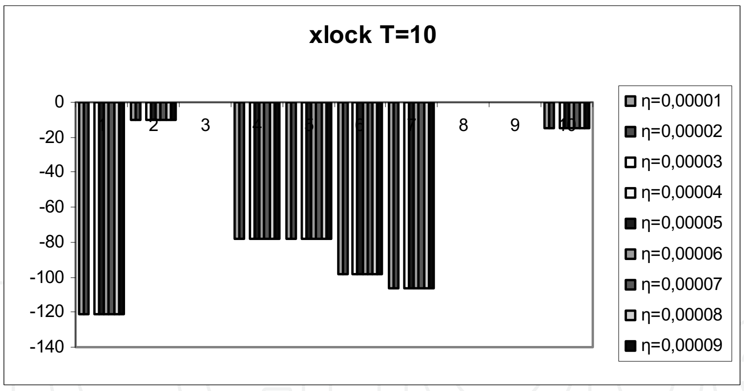

Fig. 5. Values of LLR depending on the value of η when T=10 for xlock

The results of BCJR algorithm were compared against its results over training which was performed using the gradient based method. Figures 6, 7 and 8 present the values of the *LLR*s for the process synthetic sendmail, obtained by applying the above presented two methods - BCJR algorithm and BCJR algorithm over gradient training for *T*=10, 15 and 20 and η=0,00005. From Figure 8 one could see that the second method registers O_6, O_8, O_{14}, O_{19} and O_{20} as possible attacks, while the first method registers these patterns as results of normal system work. It is worth to mention that the decoding based on BCJR algorithm is more consistent with preceding gradient training.

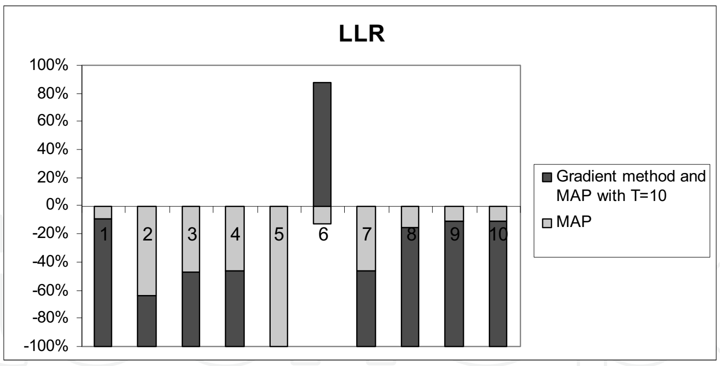

Fig. 6. The *LLR*s for the process synthetic sendmail for *T*=10

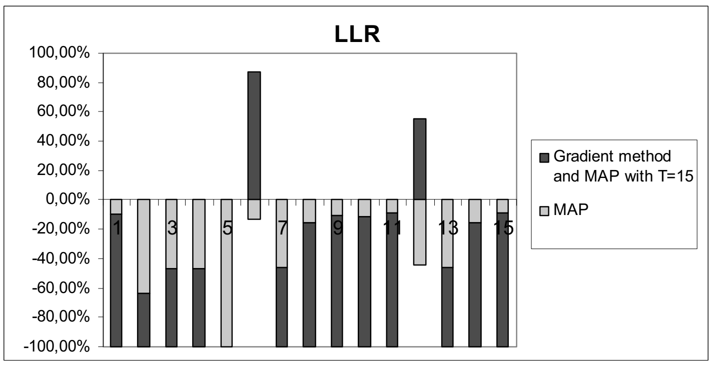

Fig. 7. The *LLR*s for the process synthetic sendmail for *T*=15

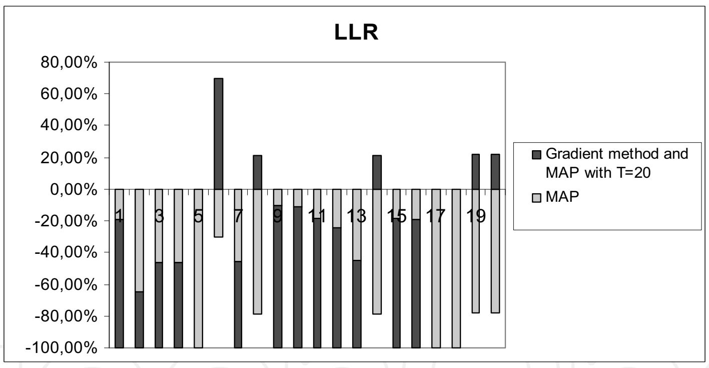

Fig. 8. The *LLR*s for the process synthetic sendmail for *T*=20

3.3 The results obtained by the Max-log MAP algorithm

The initial HMM was created according to the system states set *S*, whereupon it was tuned up using the gradient method. The result model was utilized as a normal user activity description. Then the Max log-MAP decoding algorithm was applied in order to distinguish normal activities from abnormal ones. The results of this algorithm are *LLRs*, which represent the probability of intrusion occurrence at a given moment of time. As in the previous case, the intrusion data were examined consequently with sliding window with length *T*: 10, 15 and 20. For standard gradient descent learning rate η was applied with the values from 0,00001 to 0,00009 with step 0,00001 for both observation and transition probabilities.

Figures 9, 10 and 11 contain the result values of *LLRs* in the case of *T*=10, when *η* takes the values between 0.00001 and 0.00009, for the processes synthetic ftp, named and xlock. As the Max log-MAP algorithm follows only the path that maximizes the transition probabilities, this may cause a lack of precision in the intrusion detection. In contrast, the MAP algorithm takes into account the whole trellis of possible system paths.

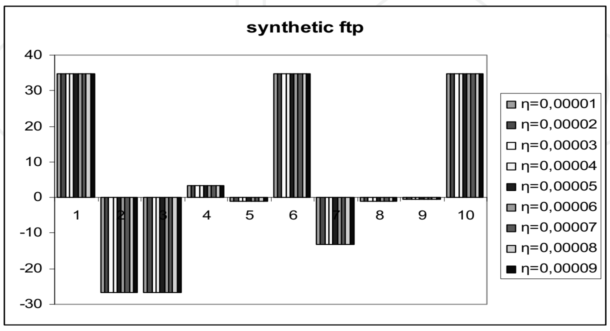

Fig. 9. Values of LLR depending on the value of η when T=10 for synthetic ftp

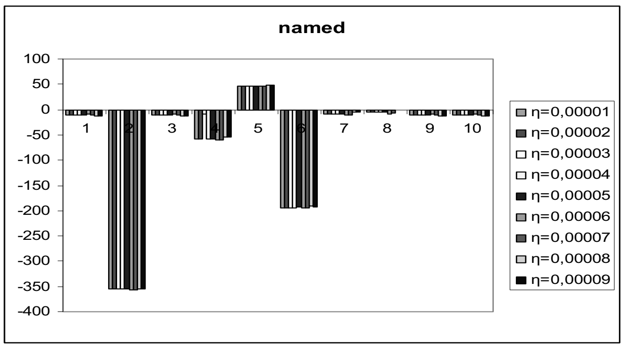

Fig. 10. Values of LLR depending on the value of η when T=10 for named

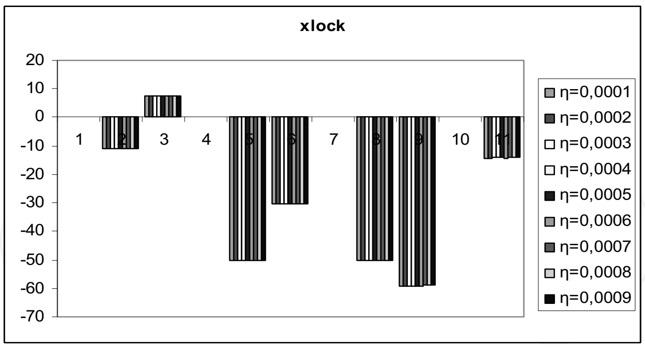

Fig. 11. Values of LLR depending on the value of η when T=10 for xlock

3.4 The results obtained by the JTA

The state transition probabilities were evaluated based on the normal user activity patterns during the system work in attack absence. A slide window with length *T*=10 was used in order to cross the traces of current user activity. The most likely states sequence and the corresponding state transition probabilities were obtained for each unknown observation sequence. Comparing the obtained state transition probabilities with the state transition probabilities of the normal user activity patterns the intrusions presence or absence was determined. Some of the results from the distribution of anomaly signal for the processes synthetic ftp, named and xlock are presented in Figures 12, 13and 14:

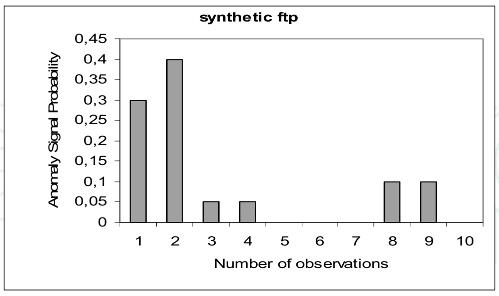

Fig. 12. The distribution of anomaly signal

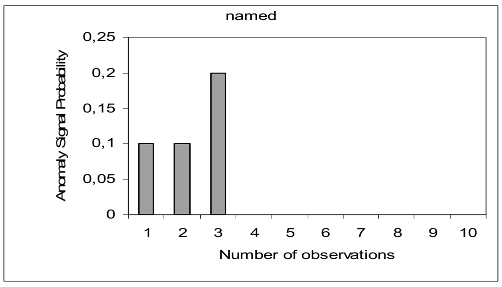

Fig. 13. The distribution of anomaly signal

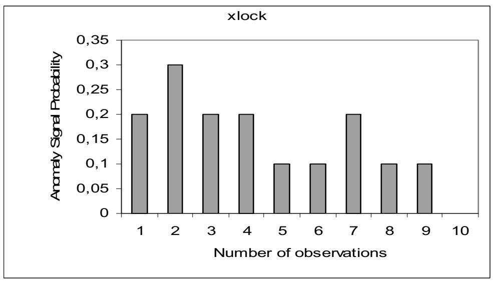

Fig. 14. The distribution of anomaly signal

4. Evaluation of the obtained results

4.1 Statistical methods of evaluating the effectiveness of IDS

The goal of the hypothesis testing is to determine whether a variation between two sample distributions can be explained by chance or not. For every possible criterion value we select to discriminate the two sets, there will be some cases with the intrusion correctly classified as positive (*TP* – True Positive), but some cases with the intrusion will be classified negative (*FN* - False Negative). On the other hand, some cases without the intrusion will be correctly classified as negative (*TN* - True Negative), but some cases without the intrusion will be classified as positive (*FP* - False Positive).

$$FP = Condition\ absent + Positive\ result$$

$$FN = Condition\ present + Negative\ result$$

As are measure of the quality of binary classification can be used the Matthews correlation coefficient (*MCC*) (Matthews, 1975):

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

MCC takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. *MCC*=+1 represents a perfect prediction, while *MCC*=-1 represents the worst possible prediction. It considers both the true positives and the true negatives as successful predictions.

The false positive rate (*FPR*) is the frequency with which the IDS reports malicious activity in error. The probability that an observed positive result is a false positive may be calculated using Bayes's theorem, whose basic concept is that the true rates of false positives and false negatives are not a function of the accuracy of the test alone, but also the actual rate or frequency of occurrence within the test set.

$$FPR = \frac{number\ of\ false\ positives}{total\ number\ of\ negative\ instances}$$

The true danger of a high *FPR* lies in fact that it may cause to ignore the system's output when legitimate alerts are raised. The false negative rate (*FNR*) is the frequency with which the IDS fails to raise an alert when malicious activity actually occurs, i.e. they represent undetected attacks on a system.

$$FNR = \frac{number\ of\ false\ negatives}{total\ number\ of\ positive\ instances}$$

FNR changes in an inverse proportion to *FPR*.

The crossover error rate (*CER*) is defined as adjusting the system's sensitivity until the *FPR* and the *FNR* are equal. In order to achieve a balance between *FPR* and *FNR*, we may select the *IDS* with the lowest *CER*.

Sensitivity is a probability that a test result will be positive when the intrusion is present (true positive rate - *TPR*).

$$FNR = 1 - Sensitivity$$

A sensitivity of 100% means that the test recognizes all intrusion as such. In the language of statistical hypothesis testing, it is called the statistical power of the test. Specificity is a probability that a test result will be negative when the intrusion is not present (true negative rate - *TNR*).

$$FPR = 1 - Specificity$$

A specificity of 100% means that the test recognizes all normal activity as normal activity.
The receiver operating characteristic (*ROC*) curve (Ferri et al., 2005, Hanley&McNeil, 1982) is a method of graphically demonstrating the relationship between sensitivity and specificity. An *ROC* space is defined by *1-specificity* and *sensitivity* as *x* and *y* respectively, which depicts relative trade-offs between *TP* and *FP*. As the decision threshold moves to the right along the *x*-axis, sensitivity ranges from one, when all tests are read as abnormal (no *FN*), to 0, when all are normal (no *TP*). Maximal sensitivity is realized when all tests are reported as abnormal. Specificity moves in concert from 0 (no *TN*) to one (no *FP*). Maximal specificity is achieved by reporting all tests as normal. The best possible prediction method would yield a point in upper left corner (0,1) of the *ROC* space, representing 100% sensitivity (all *TP* are found) and 100% specificity (no *FP* are found). This point is called a perfect classification. The diagonal line (from the left bottom to the right corner) divides the *ROC* space in areas of good and bad classification. Points above this line indicate good classification results, while points below the line indicate wrong results. Accuracy is measured by the area under the *ROC* curve. An area from 0.9 to 1 represents an excellent result; an area from 0,8 to 0,9 represents a good result, form 0,7 to 0,8 – a fair result, from 0,6 to 0,7 – a poor result and from 0.5 to 0,6 – a fail result.
The positive predictive value is the probability that the intrusion is present when the test is positive. It is applied to evaluate the usefulness of a recognizable test.

$$PPV = Positive\ Predictive\ Value = \frac{number\ of\ True\ positives}{Number\ of\ True\ positives + number\ of\ False\ positives}$$

The negative predictive value is the probability that the intrusion is not present when the test is negative.

$$NPV = Negative\ Predictive\ Value = \frac{number\ of\ True\ negatives}{number\ of\ True\ negatives + number\ of\ False\ negatives}$$

Accuracy is degree of conformity of a calculated quantity of the anomaly detection method accurately verify a given unknown sequence to be normal or anomalous (Taylor, 1999).

$$Accuracy = \frac{number\ of\ True\ positives + number\ of\ True\ negatives}{number\ of\ True\ positives + False\ positives + False\ negatives + True\ negatives}$$

An accuracy of 100% means that the test identifies all anomalous and normal activity correctly.

4.2 The effectiveness of IDS based on the BCJR decoding algorithm

In order to evaluate the *FPR* we applied a method used by Hoang (Hoang et.al., 2003). This approach is based on the assumption that as a normal traces sequence does not contain any intrusions, any reported alarms could be considered as false alarms. From the normal traces we generated a list of *n* consecutive short sequences of system calls, using a sliding window of length *T*. Then, each short sequence of the list is evaluated by the detection method to determine if it is normal or abnormal. We counted all abnormal sequences for the whole list as *m*. The *FPR* is calculated as *m/n*.

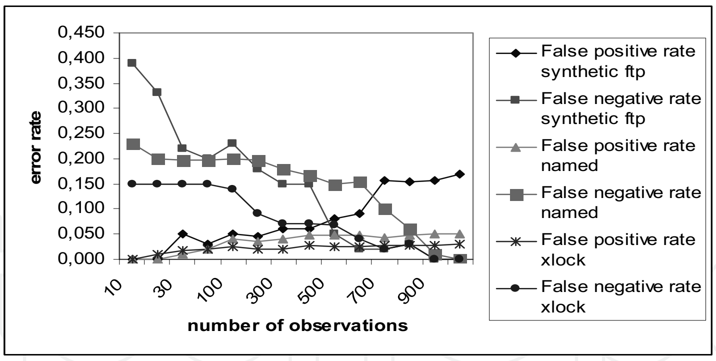

Fig. 15. The *FNR* and the *FPR* for the examined processes

Figure 15 contains graphs of the *FNR* and the *FPR* for the input processes. Table 2 contains the values of the *FPR*, *FNR*, *CER*, the accuracy and *MCC* for processes synthetic sendmail, synthetic ftp, named and xlock.

Process	*FPR*	*FNR*	*CER*	Accuracy	*MCC*
synthetic sendmail	5%	21%	0,04	83%	0,67
synthetic ftp	17%	39%	0,07	72%	0,41
named	5%	23%	0,05	86%	0,72
xlock	3%	15%	0,03	91%	0,82

Table 2. The false alarms rate and the algorithm accuracy

The proposed methodology achieves low level of the false positive rate values for the processes xlock, synthetic sendmail and named. The obtained value of 17% for synthetic ftp and the relatively high false negative rates should be further examined. As *FPR* and *FNR* are interrelated, we can reduce one at the expense of increasing the other. *CER* is the point at which the system is tuned so that both kinds of false responses occur with the same frequency. So the obtained values of *CER* could be applied as points of trade-off between *FPR* and *FNR*. Since the *CER* is the point at which these rates are equal, from Table 2 could be seen that for the examined processes the obtained values of *CER* are between 0,03 and 0,07, which implies low error and high accuracy rate of the proposed methodology.

The conducted experiments indicate the proposed methodology produces results with high level of accuracy, since all obtained values are between 72% and 91% for all examined processes. The obtained values of *MCC* are between 0.41 and 0.82, which indicate significant correlation between the current activity data and the data from the normal activity description, as the value of 1 denotes a perfect correlation.

In a *ROC* curve each sensitive value can be plotted against its corresponding specificity value to create the diagram for the examined processes in Figure 16. A methodology with perfect discrimination has a *ROC* plot that passes through the upper left corner, consequently the closer the *ROC* plot is to the upper left corner, the higher the overall accuracy of the test (Zweig&Campbell, 1993). The points in the upper left corner of the *ROC* space, which are produced by the proposed methodology for the processes synthetic ftp, named and xlock are (0,09; 0,97), (0,05; 0,98), (0,03; 0,99) respectively.

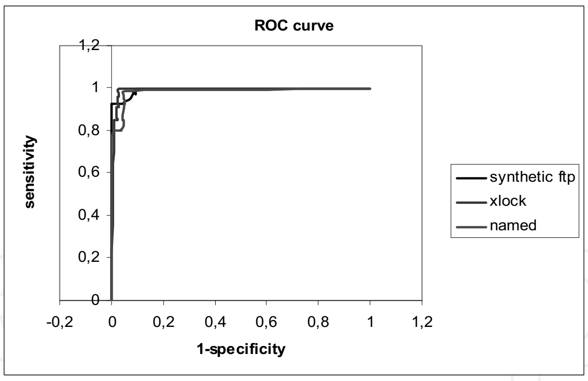

Fig. 16. The *ROC* curve for IDS based on BCJR algorithm

4.3 The effectiveness of IDS based on the Max-log MAP algorithm

In order to evaluate the effectiveness of *IDS* based on the Max-log MAP algorithm we determine the sensitivity, specificity, *PPV*, *NPV* and accuracy for the processes named, synthetic sendmail, synthetic ftp and xlock, which results are presented in Tables 3, 4, 5 and 6 respectively.

synthetic sendmail	true	false	
positive	True positive=12	False positive=125	PPV=8,7%
negative	False negative=9	True negative=1223	NPV=99,3%
	Sensitivity=57,1%	Specificity=90,7%	Accuracy=90,2%

Table 3. The sensitivity, specificity, *PPV, NPV* and accuracy for the process synthetic sendmail

synthetic ftp	true	false	
positive	True positive=8	False positive=154	PPV=4,9%
negative	False negative=6	True negative=1140	NPV=99,5%
	Sensitivity=57,1%	Specificity=88,1%	Accuracy=87,8%

Table 4. The sensitivity, specificity, *PPV, NPV* and accuracy for the process synthetic ftp

named	true	false	
positive	True positive=11	False positive=135	PPV=7,5%
negative	False negative=7	True negative=1244	NPV=99,4%
	Sensitivity=61,1%	Specificity=90,2%	Accuracy=89,8%

Table 5. The sensitivity, specificity, *PPV, NPV* and accuracy for the process named

xlock	true	false	
positive	True positive=9	False positive=104	PPV=7,9%
negative	False negative=7	True negative=1125	NPV=99,4%
	Sensitivity=56,2%	Specificity=91,5%	Accuracy=91,1%

Table 6. The sensitivity, specificity, *PPV, NPV* and accuracy for the process xlock

The *CER* and the *MCC* for the examined processes are represented in Table 7. As the smaller value of *CER*, the better the intrusion detection performance, the presented results show the proposed method gives its best results for processes named, synthetic ftp and synthetic sendmail and satisfactory value for xlock. Since the *CER* value indicates that the proportion of false acceptances is equal to the proportion of false rejections, the lower the equal error rate value is, the higher the accuracy of the proposed methodology is.

As a single measure of the performance of the test, the values of the *MCC* in Table 7 indicate that the proposed method gives feasible results during the recognition stage. The *MCC* values for all processes are between 0.44 and 0.63, which indicate significant correlation between the examined data and the data from the normal activity description.

Process	*CER*	*MCC*
synthetic ftp	0,05	0,44
synthetic sendmail	0,04	0,63
named	0,03	0,57
xlock	0,12	0,59

Table 7. The value of *CER* and *MCC* for the examined processes

The sensitivity, also referred to as recall rate, reveals how good the methodology is at correctly identifying anomalous patterns. The obtained sensitivity values, presented in Tables 3-6, indicate the proposed methodology produces good sensitivity rates, since all calculated values belong to the interval (56%, 62%). Specificity, on the other hand, is concerned with how good the methodology is at correctly identifying patterns of normal system activity. Both of them may range from 0 to +1 and the latter value is associated with perfect predictions. The proposed methodology achieves high specificity rates, as the obtained values for all processes are between 88% and 92%.

Considering the obtained predictive values, we see in Tables 3-6 that all *PPV* are between 4,9% and 8,7%, while all *NPV* are between 99,3% and 99,5%. Since the positive predictive values refers to the chance that a positive test result will be correct, the obtained results show that the proposed method correctly classifies the patterns with high degree of probability. On the other hand, negative predictive value is concerned only with negative test results. From the Tables 3-6 we see that the proposed methodology produces results with excellent negative predictive values. The both predictive values depend on the prevalence of the intrusions, since they depend on the number of true positives and false negatives and true negatives and false positives, respectively.

Since the accuracy values for all processes belong to the interval (87%, 92%), we can conclude that the proposed methodology produces precise and reliable detection results.

4.4 The effectiveness of IDS based on the *JTA*

Figure 17 contains graphs of the *FNR* and the *FPR* for the processes synthetic sendmail, synthetic ftp, named and xlock. From the graphs could be seen that the *CER* values as trade-offs between the *FPR* and the *FNR* are between 0,02 and 0,18 for the examined processes. Since the *CER* is the point at which *FPR* and *FNR* are equal, the obtained results imply low error and high accuracy rate of the proposed methodology.

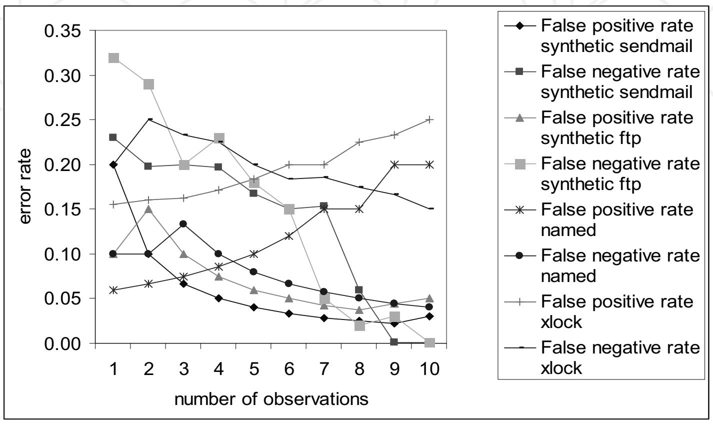

Fig. 17. The *FPR* and the *FNR* for the examined processes

A *ROC* curve is a non-parametric approach to evaluate a binary classification method. It is a two-dimensional depiction of the results, where each sensitive value can be plotted against its corresponding specificity value to create the diagram for the examined processes in Figure 18. The points in the upper left corner of the *ROC* space for the processes synthetic sendmail, synthetic ftp, named and xlock are (0,06; 0,94), (0,05; 0,99), (0,55; 0,95) and (0,25; 0,85) respectively. As the point (0, 1) denotes the perfect detection, the proposed methodology produces reliable and qualitative results while distinguishing the normal activity from abnormal one.

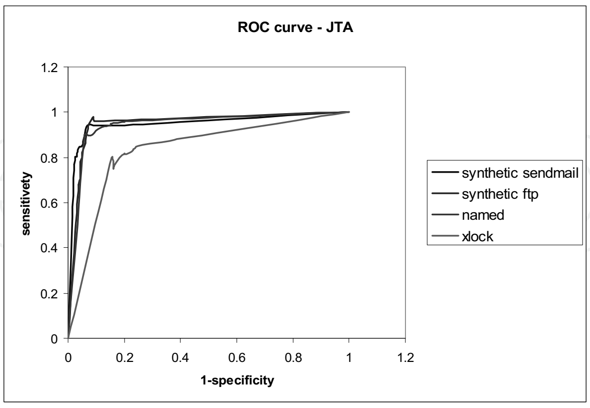

Fig. 18. *ROC* curve for IDS based on JTA

Table 8 contains the values of the *FPR*, the *FNR*, the accuracy and *MCC* for the following processes: synthetic ftp, synthetic sendmail, named and xlock.

Processes	*FPR*	*FNR*	Accuracy	*CER*	*MCC*
synthetic ftp	5%	1%	94%	0,04	0,85
synthetic sendmail	16%	6%	78%	0,02	0,60
named	11%	3%	87%	0,09	0,70
xlock	7%	2%	90%	0,18	0,69

Table 8. The false alarms rate and the algorithm accuracy

The methodology, which applies JTA during the detection stage, achieves low level of the *FPR* values for the processes synthetic ftp and xlock and low level of the *FNR* for all examined processes. The reasons of obtaining values of 16% for synthetic sendmail and 11 % for named should be further examined.

The accuracy is a key feature of each diagnostic algorithm. Analyzing the obtained accuracy values, presented in Table 8, we see that the proposed methodology produces precise and reliable results, since all accuracy values are between 78% and 94 %. The highest accuracy is achieved for the process synthetic ftp; the balanced accuracy results are obtained for the processes xlock and named; and the lowest, but still satisfactory result is obtained for the process synthetic sendmail.

One can observe that the values of *MCC*, presented in Table 8, are between 0,69 and 0,85, which reveals that there is a substantial correlation between the normal activity sequences from the created database and the observed sequences of current system activity.

4.5 Discussions

Let designate with A the methodology, based on the BCJR decoding algorithm during the detection stage, with B the methodology, based on the Max-log MAP decoding algorithm during the detection stage, and with C the methodology, based on the JTA during the detection stage. From Table 2 we see that method A achieves the highest accuracy value of 91% for the process xlock, while the lowest accuracy value of 72% is obtained for the process synthetic ftp. As one can see in Tables 3-6 method B achieves its highest accuracy value of 91,1% for the process xlock, since its lowest value of 87,8% is achieved for the process synthetic ftp. At last method C achieves its highest accuracy value of 94% for the process synthetic ftp, and reaches its lowest accuracy value of 78% for the process synthetic sendmail. Comparing all methods when accuracy alone is concerned, we can conclude the method B outperforms other methods, since it produces stable and reliable accuracy results for all examined processes.

Comparing the proposed methods when *MCC* alone is concerned, we see that method C yields the best overall correlation results, followed by methods A and B respectively. This means the overall correlation for all processes between the predicted and observed behavior is greater, when the methodology applies JTA, than the case when BCJR or Max-log MAP is applied.

The results indicate the proposed methodology with HMM, that describes the normal system activity, could lead to development of IDS with qualitative performance and high level of classification accuracy. The major drawbacks of the approach are the relatively high amount of resources necessary for the normal activity description and the relatively high false positives rate. But we have to outline the normal behavior creation is performed only once during system initialization.

5. Conclusion

The present work introduces an intrusion detection methodology that uses HMM for normal activity description and some decoding algorithms for detecting attacks targeted at essential server processes. The methodology relies on probabilistic methods for both algorithm stages: the normal activity description and the intrusion detection itself. The learning-based approach was applied in order to increase the system ability to detect novel attacks, which is among the most important features of the anomaly IDS. The feasibility of the proposed approach was justified by simulation experiments and evaluation of the obtained results.

6. References

Bahl L., J.Cocke, F.Jelinek, and J.Raviv, (1974). Optimal Decoding of Linear Codes for minimizing symbol error rate, *IEEE Transactions on Information Theory*, vol. IT-20(2), pp. 284-287.

Bahrololum M., M. Khaleghi (2008). Anomaly Intrusion Detection System Using Gaussian Mixture Model, *Third International Conference on Convergence Information Technology*, pp. 1162-1167, Busan, Nov. 2008, vol. 1, no. 1.

Benedetto S., D. Divsalar, G. Montorsi, F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," IEEE Comm. Letters, Vol. 1, No. 1, pp. 22-24, Jan. 1997.

Dagorn N. (2008). WebIDS: A Cooperative Bayesian Anomaly-Based Intrusion Detection System for Web Applications, *Recent Advances in Intrusion Detection*, LNCS, Vol. 5230/2008, Springer Berlin / Heidelberg, pp. 392-393.

Ferri C., N. Lachinche, S. A. Macskassy, A. Rakotomamonjy, eds., (2005). Second Workshop on ROC Analysis in ML, Bonn, Germany, August 2005.

Forrest S., S.A. Hofmeyr, A. Somayaji, (1998). Intrusion detection using sequences of system calls, *Journal of Computer Security*, Vol. 6, pp. 151-180.

Forrest S., S.A. Hofmeyr, A. Somayaji, T.A. Longtaff, (1996). A Sense of Self for Unix Processes, *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp.120-128, IEEE Computer Society Press, Los Alamitors, CA.

Ghosh K.A. et.al (1999). Study in Using Neural Networks for Anomaly and Misuse Detection, *Proceedings of the 8th SENIX Security Symposium*, pp 131-142, August 1999, Washington D.C.

Hanley JA, McNeil BJ (1982). The meaning and use of the area under the Receiver Operating Characteristic (ROC) curve. *Radiology*, Vol 143, pp. 29-36.

Hoang X.D., J. Hu, P. Bertok, (2003). A Multi-layer Model for Anomaly Intrusion Detection Using Program Sequences of System Calls, *11th IEEE International Conference on Networks (ICON 2003)*, Sydney, Australia.

Jecheva V., E. Nikolova, (2007). An Application of Learning Problem in Anomaly-based Intrusion Detection Systems, *Second International Conference of Availability, Reliability and Security ARES 2007*, pp. 853-860, Vienna, April 2007.

Joshi S.S., V.V. Phoha (2005), Investigating hidden Markov models capabilities in anomaly detection, *Proceedings of the 43rd ACM annual Southeast regional conference*, pp. 98 – 103, Vol. 1, Kennesaw, Georgia, USA.

Lauritzen S.L., (1996). *Graphical Models*, Oxford Science Publications.

Lauritzen, Steffen L.; Spiegelhalter, David J. (1988). Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems, *Journal of the Royal Statistical Society*, Series B (Blackwell Publishing) 50, pp.157–224.

Matthews B.W., (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim*. Biophys. Acta, 405, pp.442-451.

Nikolova E., V. Jecheva, (2007). Anomaly Based Intrusion Detection Based on the Junction Tree Algorithm, *Journal of Information Assurance and Security*, Dynamic Publishers Inc., Vol. 2, Issue 3, pp. 184-188.

Nikolova E., V. Jecheva, (2008). Some Evaluations of the Effectiveness of Anomaly Based Intrusion Detection Systems Based on the Junction Tree Algorithm, *Proceedings of the 5th CITSA 2008*, Orlando, Florida, June 29th - July 2nd, 2008, vol. 1, pp.115-120.

Qiao, Y., Xin, X.W., Bin, Y. & Ge, S.(2002) Anomaly intrusion detection method based on HMM, *IEEE Electronic Letters,* Online No: 20020467.

Rabiner L. R., B. H. Juang, (1986). An introduction to Hidden Markov Models, *IEEE ASSP Magazine*, pp.4-16.

Robertson P., E. Villebrun, P. Höher, (1995). A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, *IEEE Int. Conf. On Communications*, pp. 1009-1013, Seattle, WA, Jun. 1995.

Tan X., H. Xi (2008), Hidden semi-Markov model for anomaly detection, *Applied Mathematics and Computation*, Vol. 205, Issue 2, November 2008, Special Issue on Advanced Intelligent Computing Theory and Methodology in Applied Mathematics and Computation, pp. 562-567.

Taylor J. R., (1999). *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, pp.128-129.

University of New Mexico, Computer Immune Systems Project, http://www.cs.unm.edu/~immsec/systemcalls.htm

Vigna G., E. Jonsson, C. Kruegel, (2003). Recent Advances in Intrusion Detection, *Proceedings of 6th International Symposium, RAID 2003*, Pittsburgh, PA, USA, September 2003, Springer.

Zweig M.H., G. Campbell, (1993). Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine, *Clinical Chemistry*, Vol. 39, Num. 4, pp. 561-577.