Journal of Automation and Information Sciences, 35(6), 2003

Adaptive Anomaly Detection of Computer System User's Behavior Applying Markovian Chains with Variable Memory Length. Part I. Adaptive Model of Markovian Chains with Variable Memory Length

N.N. KUSSUL, A.M. SOKOLOV

A new method of anomaly detection in computer systems is proposed. It is based on an adaptive modification of Markovian chains model with a variable memory length. At the first part the adaptive Markovian model with variable memory length is justified.

Key words: anomaly detection, approximation properties, automaton behavior, modeling a user's behavior.

At present the work of many institutions, companies and natural persons essentially depends on reliability of operating and protection of computer systems. The degree of security of data processed ranges from private and commercial information to military and state secret. Obtaining the unauthorized access to computer system, destruction, variation or disclosure of information, illegal use of resources or bringing the system to malfunction might have undesirable consequences for owners of information or a system. Within the last decade the world made sure that even the most reliable systems with a huge amount of time and money spent on their construction are not always capable of protecting computer systems of commercial, state and military institutions from hackers' attacks.

Hence, the development of mechanisms for detecting the attempts of unauthorized access remained urgent. The intrusion detection system (IDS) is designed for this problem solution.

This paper proposes the detection method of computer system anomalies based on adaptive modification of a model of Markovian chains with variable memory length. The first part of the work justifies the adaptive model of Markovian chains with variable memory length. The second part suggests several ways of detecting anomalies via a new model as well as the method to deal with replay-attacks assuming substitution of some legitimate data.

1. IDS analysis

We divide IDS into two classes, namely: misuse detection systems to respond to the known attacks and anomaly detection systems to reveal deviation of system evolution process from the normal one.

1.1. Misuse detection systems. A computer attack almost always represents a multistage process to be performed by highly qualified hacker. Hence, the simplest way of hackering consists in employing the standard means (exploits), i.e., already written moduli to realize all the necessary attack stages on the definite sensitive areas of a system.

The operation of misuse detection systems is based on making patterns or "signatures" of attacks. This kind of protection systems are effective on the known attack schemes. However, in case of new unknown attack or deviation of attack run from pattern arise serious problems. So one has to support large database including each attack and its variations and to supplement permanently the pattern base.

ISSN 1064-2315

1.2. Anomaly detection systems. Anomaly detection systems unlike expert systems are more flexible and allow one to detect unknown attacks. Anomaly detection systems are based on assumptions that all actions of a hacker are certain to differ from a common user behavior, i.e., they are abnormal. Such system run was preceded by the period of information accumulation when the concept of normal activity of system, process or a user is being constructed. It becomes a pattern to assess the following data.

The construction of normal behavior pattern is often complicated by the absence of data which contain hacking traces. Hence training is supposed to be conducted only with the positive examples that will essentially complicate the problem. The testing of anomaly detection system is also often carried out in the absence of real data on attack "contents". At that the use is made of cross tests when the data obtained from one subject are being tested by the behavior pattern of the other.

After composing the normal behavior model the anomaly detection system fixes the necessary part of the system parameters to verify them with respect to the model correspondence. At that two forms of errors are possible.

1. The normal behavior of a system or a user are mistakenly taken as cracking (*false positives*).
2. The attempt of hacking is taken as a normal activity (*false negatives*).

With these both situation being unwanted, the second is more dangerous. One of the main construction problems of anomaly detection system is an accurate determination of conditions under which the situation is taken as abnormal so that the above errors would not occur too often.

1.3. Approach to detection of anomalies based on Markovian chains with variable memory length. One of the first works on anomaly detection was the communication [1] which determined the basic notions and approaches to this problem. Later in literature there appeared a lot of examples of constructing anomaly detection systems Most of them are conceptual models aimed at verification of possibility of employing the mathematical model or the approach.

Nearly all the described anomaly detection models can be subdivided into such categories:

- based on enumeration and storage of behavior examples [2];
- frequency ones [1];
- neuronet ones [3];
- based on construction of finite automatons [4–6].

A large data bulk accessible for audit systems in computer systems are of successive and nondeterministic nature. Moreover, many sources which generate these sequences are characterized by probability of the next signal or symbol being dependent on the previous ones. Very often they depend on a small number. These sources are said to have "short memory". This leads to the idea of modeling such successive data via Markovian chains [1, 7–9] or concealed Markovian models [4]. Although these statistic models allow one to describe a wide class of sequences they have serious limitations: the size of Markovian chains exponentially increases with growth of their degree, i.e. the number of states of the corresponding automaton behaves as $O(|\Sigma|^L)$, with $|\Sigma|$ being the size of symbol alphabet, L being the chain degree. Hence, only models with a small degree are of practical importance. However, they can badly approximate the sequences we are interested in. As for concealed Markovian models, apart from theoretically proved complexity of their training, attempts of their practical realization showed that their adjustment required great inputs.

The experience bears witness that the next symbol or signal is rarely determined by the previous context of the constant length. If one considers the sequences of users' commands, then the probability of the next command is determined by the context of variable length. Hence, there is no necessity to take into consideration all possible contexts. It is sufficient to know only those which the next command depends on. Good anomaly detection system is certain to consider that. Hence, the selected model enables one to obtain the Markovian chains with variable memory length by series of examples of a certain class of automatons running [10].

A bulk of described approaches to modeling a users' behavior, programs or systems are nonadaptive. In better case they propose periodic realignment of model aimed at adjustment to variable behavior. It consumes time and decreases reliability of models in the course of time passed since the next retraining. A single known for us work which employs method similar to our method of adaptive variation of

probabilities is communication [7]. However, it was based on the simplest Markovian chain of the first order and the context whose length exceeds one command is not taken into account.

To support the real adaptability (when for obtaining the updated model the current model is modified in view of the information entered) one suggests the procedure of variation of model parameters so that the last information would be the most weighty. At that the useful approximation properties of the initial model are preserved.

2. The initial model

2.1. Basic concepts. Consideration is given to finite alphabet of symbols (in this case — commands) Σ, the finite set of automaton states Q where each state $q \in Q$ associates with the string-mark $s \in \Sigma^*$. The empty string is marked by e.

Definition 1. The quintuple $(Q, \Sigma, \tau, \gamma, \pi)$ with $\tau : Q \times \Sigma \to Q$ being the next state transition function is called the probabilistic suffix automaton (PSA), $\gamma : Q \times \Sigma \to [0, 1]$ is the probability of the next symbol, $\pi : Q \to [0,1]$ is the initial state distribution. For the pair of states $q^1, q^2 \in Q$ and for each symbol $\sigma \in \Sigma$ if $\tau(q^1, \sigma) = q^2$ and q^1 has the mark s^1 then q^2 has the mark s^2 which is the suffix of $s^1\sigma$. Moreover, for each $q \in Q$ $\sum_{\sigma \in \Sigma} \gamma(q, \sigma) = 1$ and $\sum_{q \in Q} \pi(q) = 1$ are certain to be fulfilled.

The probability of PSA generating the sequence of symbols $r = r_1 r_2 \dots r_N$ equals

$$P_M^N(r) = \sum_{q^0 \in Q} \pi(q^0) \prod_{i=1}^{N} \gamma(q^{i-1}, r_i),$$

where $q^{i+1} = \tau(q^i, r_i)$.

PSA having nodes with the mark length no more than L is denoted by L-PSA. If the set of state marks Q coincides with Σ^L for some L (there exist all possible states with the length marks L) then such a L-PSA can be considered as Markovian chain of memory length L. Since in this case the states are easily identified by their marks automaton training is reduced to approximation of transition probabilities γ. For a general case when there exist only some states the training essentially complicates since we have to make correct identification of the current automaton state.

To solve the construction problem of automaton by the example of some PSA M running as a class of hypotheses we select the subclass of probabilistic machines — *Prediction Suffix Trees PST.*

Definition 2. The tree of degree $|\Sigma|$ with each tree edge marked by the symbol $\sigma \in \Sigma$ so that one edge with this symbol emerges from each internal node is called the prediction suffix tree T above the alphabet Σ. Each node of the tree is related with the pair (s, γ^s) where s is the string associated with "descend" starting from the tree root into the given node in the reverse order of symbols in a string which marks this node (Figure 1) and $\gamma_s : \Sigma \to [0, 1]$ — associated with s probability function of the next symbol. It is required that $\sum_{\sigma \in \Sigma} \gamma_s(\sigma) = 1$ for each string s marking the node in the tree.

Probability of PST generating the symbol sequence $r = r_1 r_2 \dots r_N$ equals

$$P_T^N(r) = \prod_{i=1}^{N} \gamma_{s^{i-1}}(r_i), \qquad (1)$$

where $s^0 = e$ and for each $1 < j < (N-1)$, s^j is the string to mark the deepest node where we get after passing along the tree according to $r_j\, r_{j-1} \dots r_1$ starting from the tree root T. For example, for the tree in Figure 1 the probability of the string 00101 equals 0,5*0,5*0,25*0,5*0,6 and the marks of nodes in which symbols are sequentially generated are of the form $s^0 = e$, $s^1 = 0$, $s^2 = 00$, $s^3 = 1$, $s^4 = 010$.

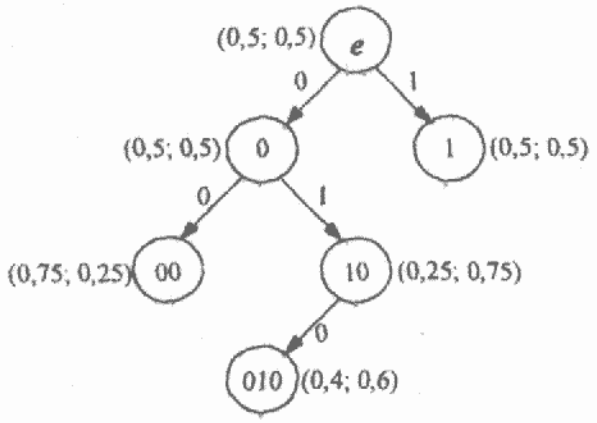

Figure 1

The following theorem is valid [10].

Theorem 1. For each L-PSA M one can construct equivalent with respect to statistic characteristics PST T_M with the maximal depth L and no more than $L\cdot|Q|$ nodes.

For assessment of constructed tree hypothesis $\hat{T}$ one needs the concept *hypothesis quality*. It is necessary that the distance between probabilistic distributions P_M and P_T of strings generated by them should not exceed the prescribed number $\varepsilon > 0$ with large probability.

Definition 3. PST $\hat{T}$ is called ε-good hypothesis with respect to PSA M if for any natural N we have

$$\frac{1}{N} D_{KL}[P_M^N \| P_T^N] \le \varepsilon,$$

where

$$D_{KL}[P_M^N \| P_T^N] = \sum_{r\in\Sigma^N} P_M^N(r) \log \frac{P_M^N(r)}{P_T^N(r)}$$

is the relative entropy.

By Theorem 1 for any PSA there exists PST equivalent to it. Conversely, it is possible to prove that for each PST one can construct "nearly" equivalent PSA (on the first L symbols probabilities of next symbols might differ). Hence, PST hypothesis $\hat{T}$ constructed in the course of algorithm operation can be transformed into the corresponding PSA hypothesis $\hat{M}$ [10].

2.2. Algorithm of PST construction. By making use of the examples of the target automaton M running we determine the *empirical* probability $\tilde{P}$. For the given subsequence of symbols s $\tilde{P}(s)$ is the relative number of enterings s into this sequence, $\tilde{P}(\sigma|s)$ is the relative number of enterings σ into the string after s, i.e., if m is the string length r and L is the maximal length s then by determining $x_j(s)$ as 1 when $r_{j-|s|+1}\dots r_j = s$ and 0 otherwise, we have

$$\tilde{P}(s) = \frac{1}{m-L+1}\sum_{j=L}^{m-1}\chi_j(s), \quad \tilde{P}(\sigma|s) = \frac{\sum_{j=L}^{m-1}\chi_{j+1}(s\sigma)}{\sum_{j=L}^{m-1}\chi_j(s)}.$$

If there exist m' strings of the length $l \ge L+1$ all together, then

$$\tilde{P}(s) = \frac{1}{m'(l-L+1)}\sum_{i=1}^{m'}\sum_{j=L}^{m-1}\chi_j(s), \quad \tilde{P}(\sigma|s) = \frac{\sum_{i=1}^{m'}\sum_{j=L}^{m-1}\chi_{j+1}(s\sigma)}{\sum_{i=1}^{m'}\sum_{j=L}^{m-1}\chi_j(s)}. \qquad (2)$$

The work of PST construction algorithm depends on such parameters: L is the maximal length of state marks, n is the upper bound of number of states in the goal PSA M as well as the parameters $\varepsilon > 0$ and $\widetilde{P}(.)$. The algorithm is required to produce PST hypothesis $\hat{T}$ which with probability $1-\sigma$ at the minimum would be ε-good hypothesis with respect to PSA M.

The algorithm starts working from the tree with one root node e. Then the tree is supplemented with the next nodes which, in our opinion, are supposed to be its belonging. Hence, the node with the mark s becomes the tree's leaf if the empiric probability $\widetilde{P}(s)$ is not negligible and for some symbol σ empiric probability $\widetilde{P}(\sigma|s)$ essentially differs from empiric probability of its obtaining after the suffix s $\widetilde{P}(\sigma|\text{suffix}(s))$, i.e., s serves the defining context for σ. The algorithm completes its operation if there is no longer the leaf for which the above conditions are valid or the restriction on the maximal depth of the tree L is attained.

The algorithm operation employs complementary set of quantities: $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$ and $\gamma_{\min}$. They are simple functions of ε, δ, n, L and $|\Sigma|$, whose specific form is determined in the course of algorithm analysis (see [10] about details).

Algorithm

1. To initialize $\hat{T}$ by one node e and the set

$$\overline{S} = \{\sigma | \sigma \in \Sigma,\ \widetilde{P}(\sigma) \geq (1-\varepsilon_1)\varepsilon_0\}.$$

2. Until $\overline{S}$ is nonempty to perform the following: to select some $s \in \overline{S}$ and

(a) to eliminate s from $\overline{S}$,

(b) if there exists $\sigma \in \Sigma$ such that

$$\widetilde{P}(\sigma|s) \geq (1+\varepsilon_2)\gamma_{\min} \tag{3}$$

and simultaneously

$$\frac{\widetilde{P}(\sigma|s)}{\widetilde{P}(\sigma|\text{suffix}(s))} \geq 1+3\varepsilon_2, \tag{4}$$

then to supplement the tree with node marked by s and (possibly) all nodes while passing from the deepest node in $\hat{T}$ which is the suffix s to the node marked by s.

(c) if $s < L$ then to supplement each $\sigma' \in \Sigma$ with the string $\sigma' s$ in $\overline{S}$ if

$$\widetilde{P}(\sigma' s) \geq (1-\varepsilon_1)\varepsilon_0.$$

The work [10] has proved the following basic theorem

Theorem 2. For any PSA M and parameters $\varepsilon > 0$, $0 < \delta < 1$ the algorithm constructs PST $\hat{T}$ such that with probability $1-\delta$ at the minimum the tree $\hat{T}$ is ε good hypothesis with respect to M. If the algorithm has access to the source of independently generated strings then the time of its operation is restricted by the polynomial of $L, n, |\Sigma|, 1/\varepsilon, 1/\delta$.

To prove this theorem the use is made of complementary lemma.

Lemma 1. There exists the polynomial m_0 such that the set of examples from $m > m_0$ (L, n, $|\Sigma|$, $1/\delta$, ε_0, ε_1, ε_2) strings with the length more than $(L+1)$ each is typical with probability $1-\delta$ at most.

Informally speaking the set of strings is considered typical if for each generated M substring of symbols its empiric probability and probability of the next symbol under condition of this substring are close to their mathematical expectations.

Definition 4. The set of strings is called typical if for each $s \in \Sigma^{\leq L}$ the following is fulfilled:

1) if $s \in Q$, then $|\widetilde{P}(s) - P(s)| \leq \varepsilon_1 \varepsilon_2$;

2) if $P(s) \geq (1-\varepsilon_1)\varepsilon_0$, then $\forall \sigma \in \Sigma, |\widetilde{P}(\sigma|s) - P(\sigma|s)| \leq \gamma_{\min}\varepsilon_2$.

It is worth noting that in definition of empiric probabilities and correspondingly in Definition 4 probability $\widetilde{P}(s)$ in case of some strings is the sum of bounded random quantities. To determine $\widetilde{P}(s)$ one needs its arithmetic mean, i.e., all terms are added with the same weighting coefficients $\frac{1}{m'(l-L+1)}$. It enables one to apply the known inequalities to sums of independent random quantities.

So for the proof of Lemma1 the use is made of Heoffding inequality variant [11] which produces the upper estimate of probability that the sum of independent bounded random quantities does not exceed its mathematical expectation.

Theorem 3. Let $X_1, X_2, ..., X_n$ be independent bounded random quantities, $S = \sum_{i=1}^{n} X_i$. Then for any $\varepsilon > 0$ we have

$$P\{|S - MS| \leq n\varepsilon\} \geq 1 - 2e^{2n\varepsilon^2}.$$

3. The model modification and PST construction algorithm

3.1. The model modification. The feature of our problem is the necessity of adaptive construction and modification of suffix tree. To provide a model with a certain adaptivity we propose to modify the algorithm of computing empiric probabilities so that the contribution of earlier examples into the total empiric probability would decrease with every step. Then later examples will be taken with larger weighting coefficients to model "forgetting" of earlier examples. Correspondingly, the algorithm of tree construction is required to be modified.

Given the definite training coefficient $0 < \alpha < 1$, we put empiric state probability with the mark s at the time instant t

$$\widetilde{P}_1(s) = P'_1(s),$$

$$\widetilde{P}_t(s) = \alpha \widetilde{P}_{t-1}(s) + (1-\alpha)P'_t(s) = \alpha^{t-1}P'_1(s) + (1-\alpha)\sum_{\tau=2}^{t}\alpha^{t-\tau}P'_\tau(s), \quad (5)$$

where $P'_t(s)$ is the empiric state probability in the string that entered at the time instant t.

Probability calculation by such rules enables us to decrease impact of earlier examples and simultaneously consider newly obtained examples with larger weight The quantity of the coefficient α regulates the speed of forgetting: values close to zero will lead to fast forgetting and values close to unity — to slow one.

In our case $\widetilde{P}_t(s)$ is the sum of independent bounded random quantities $P'_t(s)$ with variable coefficients.

We assume that at the time instant $t = 1$ probabilistic suffix tree with whose operation as an example we are trying to construct approximation has varied, i.e., distribution of state probabilities has changed very rapidly. Suppose that in our object domain the situation is modeled when a user (process) varies its behavior by itself or we start observing the behavior of other subject of the system identified as a legitimate user (process). The modified algorithm should adjust to a new behavior to ensure the necessary adaptability of the model, the results of Theorem 2 being correct.

Let X_t be the empiric probabilities on the step t,

$$S_t = \alpha^t X_0 + (1-\alpha)\sum_{\tau=1}^{t}\alpha^{t-\tau}X_\tau.$$

Theorem 4. For any $\varepsilon > 0$ and $0 < \delta < 1$ there exist $0 < \alpha < 1$ and $t \geq 1$ such that in case of stepwise character of variation of quantity distribution X at the time instant $t = 1$

$$P\{|S_t - MX_t| \leq \varepsilon\} \geq 1 - \delta. \quad (6)$$

Proof mainly follows the proof scheme of Theorem 3 from [11].

From Chebyshev inequality for any $h \geq 0$ and in view of the exponent being monotonic and nonnegative function we have

$$P\{S_t - MX_t \geq \varepsilon\} \leq \frac{M e^{h(S_t - MX_t)}}{e^{h\varepsilon}}. \tag{7}$$

Since $MX_t \equiv \alpha^t MX_t + (1-\alpha)\sum_{\tau=1}^{t} \alpha^{t-\tau} MX_t$, then

$$S_t - MX_t = \alpha^t (X_0 - MX_t) + (1-\alpha)\sum_{\tau=1}^{t} \alpha^{t-\tau}(X_\tau - MX_t). \tag{8}$$

With e^{hX} being the convex function, for the bounded random value X, $0 \leq X \leq b$, we have

$$e^{hX} \leq \frac{b-X}{b} + \frac{X}{b} e^{hb}.$$

Hence,

$$Me^{hX} \leq \frac{b - MX}{b} + \frac{MX}{b} e^{hb}. \tag{9}$$

Then from (8) and (9) in view of $b = 1$ we have

$$Me^{h(S_t - MX_t)} = e^{h\alpha^t (X_0 - MX_t)} \prod_{\tau=1}^{t} e^{h(1-\alpha)\alpha^{t-\tau}(X_\tau - MX_t)} \leq$$

$$\leq e^{-h\alpha^t MX_t}(1 - MX_0 + MX_0 e^{h\alpha^t}) \prod_{\tau=1}^{t} e^{-h(1-\alpha)\alpha^{t-\tau} MX_t}(1 - MX_\tau + MX_\tau e^{h\alpha^{t-\tau}(1-\alpha)}).$$

By denoting MX_t by μ_t, $i = 0,\ldots, t$, and $b_0 = \alpha^t, b_\tau = (1-\alpha)\alpha^{t-\tau}$, $\tau = 1,\ldots, t$, we obtain

$$Me^{h(S_t - \mu_t)} \leq \prod_{\tau=0}^{t} e^{-hb_\tau \mu_t}(1 - \mu_\tau + \mu_\tau e^{hb_\tau}). \tag{10}$$

We now consider one multiplier from the product (10). Suppose

$$L(h_\tau) = -h_\tau \mu_t + \ln(1 - \mu_\tau + \mu_\tau e^{h_\tau}),$$

where $h_\tau = hb_\tau$.

We now calculate two first derivatives from $L(h_\tau)$:

$$L'(0) = -\mu_t + \mu_\tau,$$

$$L''(h_\tau) = \frac{\mu_\tau(1-\mu_\tau)e^{h_\tau}}{(1 - \mu_\tau + \mu_\tau e^{h_\tau})^2} \leq 1/4.$$

Then by expanding the function $L(h_\tau)$ into Taylor series we have

$$L(h_\tau) = L(0) + L'(0)h_\tau + \frac{L''(\xi)h_\tau^2}{2} \leq h_\tau(\mu_\tau - \mu_t) + \frac{h_\tau^2}{8}, \tag{11}$$

where $\xi \in [0, h_\tau]$.

Hence, taking into account (11) and inequalities (7) and (10) we obtain

$$P\{S_t - MX_t \geq \varepsilon\} \leq e^{-h\varepsilon} \prod_{\tau=0}^{t} e^{hb_\tau\left(\mu_\tau - \mu_t + \frac{hb_\tau}{8}\right)}.$$

The right-hand side of this expression attains minimum with respect to h at $h = 4(\varepsilon + \Delta\mu_t)/\sum_{\tau=0}^{t} b_\tau^2$ where $\Delta\mu_t = \mu_t - \sum_{\tau=0}^{t} b_\tau \mu_\tau$. Hence,

$$P\{S_t - MX_t \geq \varepsilon\} \leq \exp\left(-h\varepsilon - h\left(\mu_t - \sum_{\tau=0}^{t} b_\tau \mu_\tau\right) + \frac{1}{8}h^2 \sum_{\tau=0}^{t} b_\tau^2\right) =$$

$$= \exp\left(-\frac{2(\varepsilon + \Delta\mu_t)^2}{\sum_{\tau=0}^{t} b_\tau^2}\right). \qquad (12)$$

The fact that $P\{|S_t - MX_t| \geq \varepsilon\} = P\{S_t - MX_t \geq \varepsilon\} + P\{-S_t + MX_t \geq \varepsilon\}$ implies the estimate

$$P\{|S_t - MX_t| \geq \varepsilon\} \leq \exp\left(-\frac{2(\varepsilon - \Delta\mu_t)^2}{\sum_{\tau=0}^{t} b_\tau^2}\right) + \exp\left(-\frac{2(\varepsilon + \Delta\mu_t)^2}{\sum_{\tau=0}^{t} b_\tau^2}\right).$$

By calculating the sum $\sum_{\tau=0}^{t} b_\tau^2$ which equals $\frac{1-\alpha+2\alpha^{2t+1}}{1+\alpha}$ we obtain (in view of $\mu_t = \mu$) at $t \geq 1$

$$P\{|S_t - MX_t| \geq \varepsilon\} \leq \exp\left(-\frac{2(1+\alpha)(\varepsilon - \alpha^t(\mu_0 - \mu))^2}{1-\alpha+2\alpha^{2t+1}}\right) +$$

$$+ \exp\left(-\frac{2(1+\alpha)(\varepsilon + \alpha^t(\mu_0 - \mu))^2}{1-\alpha+2\alpha^{2t+1}}\right). \qquad (13)$$

Each term in (13) asymptotically tends to $\exp\left(-\frac{2(1+\alpha)\varepsilon^2}{1-\alpha}\right)$. Hence, from expression

$$\exp\left(-\frac{2(1+\alpha_0)\varepsilon^2}{1-\alpha_0}\right) = \frac{\delta}{2}$$

we obtain that at any $\alpha > \alpha_0 = \ln\left(\frac{2}{\delta}\right) - \varepsilon^2 \Big/ \ln\left(\frac{2}{\delta}\right) + \varepsilon^2$ for each term there exist t_1, t_2 such that at $t > \max(t_1, t_2)$ each term is bounded from above by the value $\frac{\delta}{2}$. Then their sum is smaller than δ. Whence we obtain (6).

Theorem has been proved.

3.2. Modified algorithm of PST construction. Due to altered rule of modifying empiric probabilities (5) we have to vary the algorithm of PST construction so that it would reflect those alterations.

Modified algorithm

1. To obtain $\hat{T}_{t-1}$ and the current session-example r.
2. To modify empiric probabilities of all nodes of the tree according to the rule (5).
3. To eliminate recursively all leaves from the tree of marks s for which

$$\widetilde{P}_t(\sigma) < (1-\varepsilon_1)\varepsilon_0 . \tag{14}$$

4. To initialize the set $\overline{S}$:

$$\overline{S} = \{s \mid s \in \Sigma^*, \text{suffix}\,(s) \in L(\hat{T}_{t-1}),\ \widetilde{P}_t(\sigma) \geq (1-\varepsilon_1)\varepsilon_0\},$$

where $L(\hat{T}_{t-1})$ is the set of leaves of the tree $\hat{T}_{t-1}$.

5. Until $\overline{S}$ is nonempty to perform the following: to select any string $s \in \overline{S}$ and

(a) to eliminate s from $\overline{S}$;

(b) if there exists $\sigma \in \Sigma$ such that

$$\widetilde{P}_t(\sigma \mid s) \geq (1+\varepsilon_2)\gamma_{\min} \tag{15}$$

and simultaneously

$$\frac{\widetilde{P}_t(\sigma \mid s)}{\widetilde{P}_t(\sigma \mid \text{suffix}\,(s))} > 1+3\varepsilon_2 \tag{16}$$

to supplement the tree with the node corresponding s (and possibly) all nodes on the path from the deepest node in $\overline{S}$ which is the suffix s to the node marked by s;

(c) if $|s| < L$, then to supplement each $\sigma' \in \Sigma$ with the string marked $\sigma' s$ in $\overline{S}$, if $\widetilde{P}(\sigma' s) \geq (1-\varepsilon_1)\varepsilon_0$.

The modified algorithm adaptively reconstructs the tree according to altered probability states. At that, states which by modified values of probabilities became neglectful (see (14) are eliminated, The states for which the conditions (15) and (16) began to fulfill are added.

References

1. Denning D.E., An intrusion-detection model, *Proceedings IEEE Symposium on security and privacy*, 1986, 118–131.
2. Forrest S., Hofmeyr S.A., Somayaji A., Longstaff T.A., A sense of self for Unix processes, *Proceedings IEEE Symposium on research in security and privacy, IEEE Computer Society Press*, 1996, 120–128.
3. Reznik A.M., Kussul N.N., Sokolov A.M., Neuronet identification of computer system users' behavior, *Kibernetika i vychislitelnaya tekhnika*, 1999, **123**, 70–79.
4. Lane T., Hidden Markovian models for human/computer interface modeling, *IJCAI-99 Workshop on learning about users*, 1999, 35–44.
5. Michael C.C., Ghosh A., Two state-based approaches to program-based anomaly detection, *ACM Transactions on information and system security*, 2002, **5** (2), 30–34.
6. Wagner D., Dean R., Intrusion detection via static analysis, Ed. by F.M. Titsworth, *Proceedings 2001 IEEE Symposium on security and privacy*, 2001, 156–169.
7. Davison B.D., Hirsh H., Predicting sequences of user actions, Predicting the Future: AI Approaches to time-series problems, 1998, 5–12.
8. Eskin E., Anomaly detection over noisy data using learned probability distribution, *Proceedings 17 Intern. conf. on machine learning*, Morgan Kaufman, San Francisco, CA, 2000, 255–262.
9. Ye N., A Markov chain model of temporal behavior for anomaly detection, *Proceedings 2000 IEEE systems, man and cybernetics, information assurance and security workshop*, 2000, 22–26.
10. Ron D., Singer Y., Tishby N., The power of amnesia: learning probabilistic automata with variable memory length, *Machine Learning*, 1996, **25** (2–3), 117–149.
11. Hoeffding W. Probability inequalities for sums of bounded random variables, *J. American Stat. Associat.*, 1963, **58** (301), 13–31.