УДК 004.056.57

Математическая модель системы распознавания разрушающих программных средств на основе скрытых марковских моделей

А. В. Козачок

Данная статья посвящена рассмотрению системы распознавания разрушающих программных средств на основе скрытых марковских моделей, подробно рассмотрена математическая модель системы распознавания и приведены результаты оценки её эффективности.

Ключевые слова: антивирусная защита, разрушающие программные средства, скрытые марковские модели, кластеризация.

Обеспечение информационной безопасности компьютерных систем в настоящее время является важнейшей задачей при построении и эксплуатации автоматизированных систем. При этом одной из угроз является заражение ЭВМ разрушающими программными средствами (РПС), направленными на модификацию (удаление) данных пользователя, кражу его конфиденциальной информации, затруднение функционирования или выведение из строя операционной системы.

В настоящее время существуют различные механизмы обнаружения РПС, среди них наиболее распространёнными являются следующие:

– сигнатурный поиск (выявление по последовательности байт вредоносного кода, однозначно характеризующего его);

– эвристический поиск (выявление по некоторым косвенным признакам программного кода, характеризующего его как вредоносного);

– «проактивные» механизмы, выявляющие выполнение процессами запрещённых операций (например, обращение в критические области памяти или внедрение исполняемого кода в другие процессы);

– поведенческие блокираторы (выявление аномального поведения пользователя на основе его профиля).

Все вышеперечисленные механизмы обладают существенными недостатками: ограничены возможности распознавания новых и модифицированных вирусов или необходимо вовлечение пользователя в процесс принятия решений о принадлежности файла к определённому классу.

Развитием эвристических механизмов распознавания РПС является структурный стохастический способ [1]. Суть данного подхода заключается в моделировании процесса порождения машинного кода различных классов программ (в частности, незаражённых файлов и файловых вирусов) с помощью стохастических грамматик. При этом были заданы два формальных языка на основе грамматик G_1 и G_2: $L(G_1)$, порождающий незаражённые разрушающим программным кодом («чистые») файлы, и язык $L(G_2)$, способный генерировать цепочки машинных команд, характерные для файловых вирусов. Распознавание заключалось в процедуре оценки вероятности порождения исследуемой, выделенной из анализируемого файла, последовательности машинных команд каждой грамматикой, и по критерию макси-

мального правдоподобия принималось решение о принадлежности файла к определённому (наиболее вероятному) классу.

Программная реализация механизма стохастического структурного распознавания РПС позволила провести экспериментальные исследования эффективности распознавания, а также выявить основные его недостатки. На рис. 1 представлены зависимости оценок вероятностей ошибок первого и второго рода от длины анализируемой последовательности при распознавании [1].

Из анализа графика видно, что основным недостатком данного подхода является высокая вероятность ложных срабатываний, равная 0.05 (при длине анализируемой последовательности 200 машинных команд), обусловленная усреднённостью класса незаражённых файлов (вероятность ошибки второго рода при этом составила 0.03). Также к недостаткам следует отнести невозможность распознавания принципиально нового машинного кода (новых классов или подклассов вирусов) и необходимость определения в процессе обучения выборок файлов, относящихся к строго определённому классу (например, файловые паразитические вирусы, текстовые редакторы, шифраторы файлов, аудиокодеки и др.), что в реальных условиях практически невозможно.

Одним из путей повышения эффективности структурного механизма распознавания РПС является введение процедуры внутриклассовой кластеризации цепочек машинных команд [2], которая позволит избавиться от усреднённости классов машинного кода. В этом случае модели кластеров будут описывать более тонкие структуры, т. е. файлы со схожими свойствами, что приведёт к снижению значений вероятностей ошибок первого и второго рода.

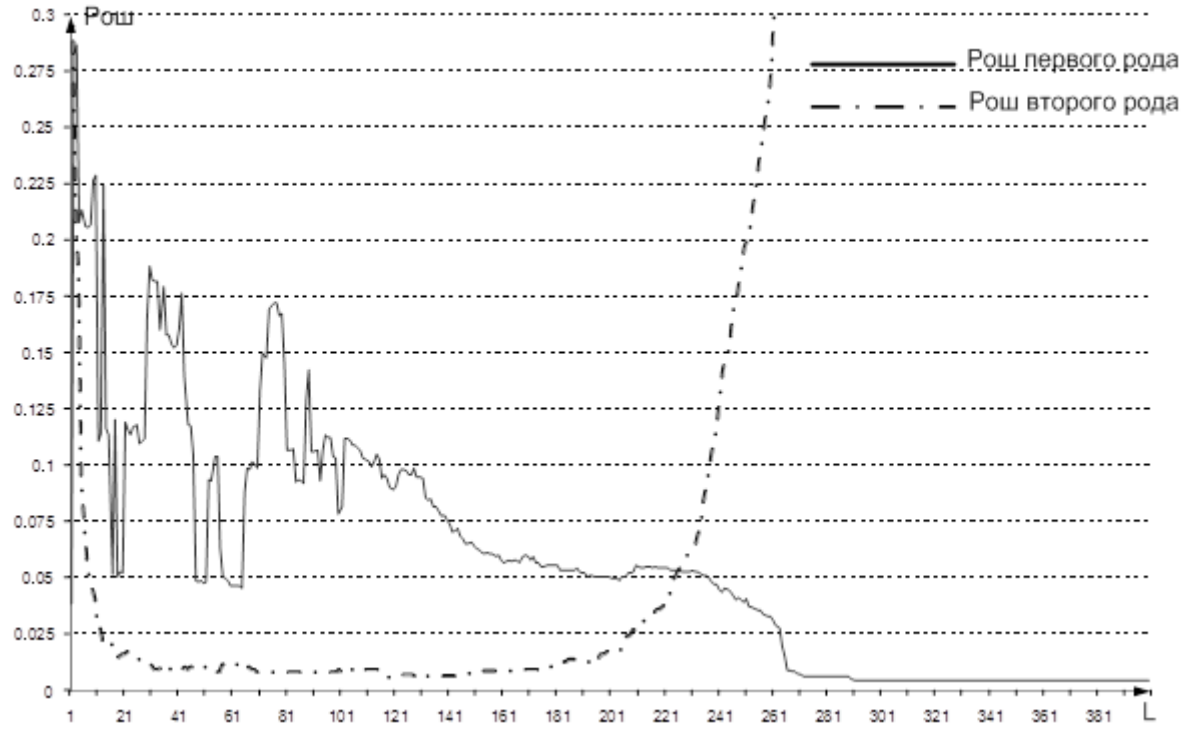

Рис.1 Зависимость ошибки первого рода и второго рода от длины анализируемой последовательности машинных команд при распознавании файловых вирусов

Для модификации стохастического структурного механизма распознавания РПС предложен механизм распознавания РПС на основе скрытых марковских моделей, который позволяет решить следующие задачи:

1) произвести автоматическую внутриклассовую кластеризацию цепочек машинных команд, выделенных из файлов обучающих выборок соответствующего класса;

2) сформировать модели каждого кластера программного кода в процессе обучения;

3) разработать процедуру распознавания класса машинного кода, к которому принадлежит исследуемый файл.

Разработка математической модели распознавания РПС на основе скрытых марковских моделей (СММ) связана с решением двух взаимосвязанных задач:

1) обучения множества СММ, задающих кластера машинного кода заражённых и незаражённых файлов (подсистема обучения);

2) разработки процедуры расчёта правдоподобия порождения данными СММ некоторой последовательности машинных команд, выделенной из анализируемого файла, и принятия решения о принадлежности данного файла к определённому кластеру (подсистема распознавания). Значение правдоподобия характеризует возможность порождения анализируемых данных исследуемой моделью.

Для обучения системы необходимо сформировать выборки исполняемых файлов для двух классов машинного кода: незаражённых файлов и РПС. Выборки при этом должны удовлетворять требованию репрезентативности, так как от них зависит эффективность работы системы распознавания. Под контрольными выборками понимаются анализируемые системой исполняемые файлы. Процедура обучения будет рассмотрена на примере одного класса незаражённых файлов, аналогичные шаги необходимо будет предпринять и над выборкой класса РПС.

Исходными данными для подсистемы обучения является множество цепочек машинных команд (ЦМК) $X = \{x_i\},\ i = \overline{1, N_{mk}}$, где N_{mk} – число ЦМК, содержащихся в репрезентативной обучающей выборке класса незаражённых файлов. ЦМК представляет собой последовательность машинных команд, выделенную из исполняемого файла путём дизассемблирования всех его экспортируемых функций. Длина выделяемой последовательности для различных исполняемых файлов различна. Под машинными командами понимается множество команд языка Ассемблера, определяемых системой команд процессоров архитектуры IA32. Система команд для данных процессоров определяется набором:

– базовых команд (например add, mov, call, jmp и др.);

– команд сопроцессора FPU (например fadd, fdiv и др.);

– специфических команд, предназначенных для оптимизации мультимедийных приложений (набор команд MMX, SSE, SEE2, 3dNow!).

Было выделено 1500 различных форм машинных команд (в данное множество входят наборы специфических машинных команд, таких как MMX, SSE2, SSE3).

Решение задачи построения подсистемы обучения включает в себя три основных этапа:

На первом этапе производится обучение множества скрытых марковских моделей по алгоритму Баума – Уэлча. Скрытая марковская модель (СММ) представляет собой дважды стохастический процесс, состоящий из пары случайных процессов: основного и ненаблюдаемого (скрытого). В данном случае основным процессом является процесс появления символов наблюдения (машинных команд), а скрытым – процесс изменения состояния модели.

Для задания СММ необходимо ввести следующие обозначения [3]:

1) N, число состояний модели. Множество состояний модели обозначим как $S = \{S_i\},\ i = \overline{1, N}$, и состояние модели в момент t – q_t.

2) M, число различных символов наблюдения, которые могут порождаться моделью (размер дискретного алфавита). Для множества наблюдаемых символов будет принято следующее обозначение: $V = \{v_i\}, i = \overline{1, M}$.

3) Распределение вероятностей переходов между состояниями (матрица переходных вероятностей) A_{ij}, где $a_{ij} = P\left[q_{t+1} = S_j \mid q_t = S_i\right],\ 1 \le i, j \le N$ – вероятность перехода модели из состояния $q_t = S_i$ в момент времени t в состояние $q_{t+1} = S_j$ в следующий момент времени $t+1$.

4) Распределение вероятностей появления символов наблюдения в состоянии j, $B=\{b_j(k)\}$, где $b_j(k)=P\left[v_k \text{ at } t \middle| q_t = S_j\right]$, где $1 \le j \le N,\ 1 \le k \le M$.

5) Начальное распределение вероятностей состояний $\pi=\{\pi_i\}$, $\pi_i = P[q_1 = S_i]$, где $i=\overline{1,N}$.

6) Последовательность наблюдаемых значений $O=O_1O_2\ldots O_T$, где O_t – символ алфавита V, T – число символов в наблюдаемой последовательности.

Скрытая марковская модель задается как $\lambda=(N,M,A,B,\pi)$, в краткой записи – $\lambda=(A,B,\pi)$. В данном случае V – дискретный алфавит, состоящий из идентификаторов машинных команд, $M=1500$. Число скрытых состояний N является одним из варьируемых параметров модели.

Обучение СММ является решением третьей общеизвестной задачи для СММ: каким образом нужно подстроить параметры модели $\lambda=(A,B,\pi)$, для того чтобы максимизировать $P(O|\lambda)$. Обучение СММ производится по алгоритму Баума – Уэлча (БУ), являющегося модификацией алгоритма обучения Expectation Modification (EM). Он предназначен для решения задачи фильтрации в аспекте СММ (поиск модели максимального правдоподобия). Данный алгоритм позволяет подобрать такие значения параметров A,B,π, при которых максимизируется правдоподобие порождения анализируемой цепочки данной моделью. Алгоритм БУ представляет собой итеративную процедуру поиска локального максимума (градиентный подъём).

Вспомогательная функция Баума позволяет проинтерпретировать алгоритм БУ как реализацию статистического алгоритма EM, в которой шаг E (вычисление математического ожидания) – это вычисление функции (1), а шаг M (модификация) – это максимизация этой функции по $\bar{\lambda}$

$$Q(\lambda,\bar{\lambda})=\sum_Q P(Q|O,\lambda)\log\left[P(O,Q|\bar{\lambda})\right]. \quad (1)$$

Вспомогательной процедурой для алгоритма БУ является алгоритм прямого – обратного хода [3]. Прямая переменная

$$\alpha_t(i)=P(O_1O_2\ldots O_t,\ q_t=S_i|\lambda)$$

представляет собой вероятность появления для данной модели λ частичной последовательности наблюдений $O_1O_2\ldots O_t$ (до момента t и состояния в этот момент). По индукции $\alpha_t(i)$ вычисляется следующим образом:

1) инициализация

$$a_1(j)=\pi_j b_j(O_1),\ 1 \le j \le N;$$

2) индуктивный переход

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)\cdot a_{ij}\right]\cdot b_j(O_{t+1}),\ 1 \le j \le N, 1 \le t \le T-1;$$

3) окончание

$$P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i). \quad (2)$$

Выражение (2) позволяет рассчитать правдоподобие, вероятность порождения моделью λ последовательности наблюдаемых переменных O.

Обратная переменная $\beta_t(i)$, определяется выражением $\beta_t(i) = P(O_{t+1}O_{t+2}\ldots O_T | q_t = S_i, \lambda)$, которое для заданной модели λ представляет собой совместную вероятность появления частичной последовательности наблюдений от момента времени $t+1$ до T и состояния S_i в момент времени t. По индукции $\beta_t(i)$ вычисляется следующим образом:

1) инициализация

$$\beta_T(i) = 1,\ 1 \le i \le N;$$

2) индуктивный переход

$$\beta_t(j) = \sum_{j=1}^{N} a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j),\ 1 \le i \le N,\ t = T-1, T-2, \ldots, 1.$$

Для описания процедуры переоценки (последовательного обновления данных и соответствующей подстройки параметров) параметров СММ вводится величина $\xi_t(i, j)$ – вероятность того, что при заданной последовательности наблюдений в моменты времени t и $t+1$ система будет соответственно находиться в состояниях S_i и S_j:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda). \tag{3}$$

Используя определения прямой и обратной переменных, выражение (3) для $\xi_t(i, j)$ примет вид

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \cdot \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \cdot \beta_{t+1}(j)}.$$

Также вводится переменная $\gamma_t(i) = P(q_t = S_i | O, \lambda)$, которая представляет собой вероятность пребывания в момент t в состоянии S_i при заданной последовательности наблюдений O и модели λ.

Используя определения прямой и обратной переменных, выражение для $\gamma_t(i)$ примет вид

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} = \sum_{j=1}^{N} \xi_t(i, j).$$

Согласно алгоритму БУ, формулы переоценки параметров СММ примут вид:

$$\overline{\pi_i} = \gamma_1(i), \tag{4}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{5}$$

$$\overline{b_j}(k) = \frac{\sum_{t=1}^{T} \gamma_t(i) | O_t = v_k}{\sum_{t=1}^{T-1} \gamma_t(i)}. \tag{6}$$

Выражение (4) показывает ожидаемую частоту пребывания в состоянии S_i в момент времени $t = 1$. Выражение (5) есть отношение ожидаемого числа переходов из состояния S_i

в состояние S_j. Выражение (6) рассчитывается как отношение ожидаемого числа переходов в состояние j и наблюдения символа v_k к ожидаемому числу переходов в состояние j.

Пусть $\lambda=(A,B,\pi)$ – исходную модель, а $\overline{\lambda}=(\overline{A},\overline{B},\overline{\pi})$ – модель после переоценки параметров. В этом случае либо 1) исходная модель определяет точку экстремума функции правдоподобия ($\lambda=\overline{\lambda}$), либо 2) модель $\overline{\lambda}$ более правдоподобна, чем модель λ, в том смысле, что $P(O|\overline{\lambda})>P(O|\lambda)$, вероятность появления последовательности O для модели $\overline{\lambda}$ больше, чем для исходной модели λ.

Итеративно повторяя процедуру переоценки параметров СММ, используя на каждом новом шаге значения параметров модели, полученные на предыдущем шаге, будут получаться модели, для которых вероятность появления последовательности наблюдений O будет увеличиваться. Процедура продолжается до тех пор, пока разность $P(O|\overline{\lambda})-P(O|\lambda)$ не станет меньше некоторого порогового значения. Глобальная сходимость итераций по алгоритму БУ доказана в [3].

В приложении к системе распознавания РПС изначально для каждой ЦМК, выделенной из обучающей выборки, O_l задаётся СММ,

$$\lambda_l=(A,B,\pi),\ l=\overline{1,N_{\text{mk}}}$$

матрицы распределений вероятностей A,B,π при этом для каждой модели заполняются равновероятными значениями (шаг инициализации алгоритма). Затем по алгоритму БУ каждая модель λ_l обучается для соответствующей последовательности наблюдаемых значений O_l, итеративно с некоторым заданным исследователем порогом точности ε.

Результатом данного этапа являются N_{mk} кластеров, обученных на последовательностях наблюдаемых значений, выделенных из репрезентативной обучающей выборки. Каждая СММ является кластером, восстановленным на основе одной ЦМК.

На втором этапе производится расчёт матрицы межкластерных расстояний. Исходя из определения СММ [3] для расчёта расстояний между моделями можно воспользоваться мерами для определения близости распределений вероятностей. Для построения матрицы межкластерных расстояний было выбрано расстояние Кульбака – Лейблера (относительная энтропия) [3].

Данное расстояние является мерой того, насколько далеки друг от друга два вероятностных распределения. Однако оно не является метрикой на пространстве распределений вследствие несимметричности.

Для дискретных распределений формула расчёта расстояния КЛ имеет вид, представленный в выражении (7),

$$D_{\text{KL}}(p,q)=\sum_X p(x)ln\frac{p(x)}{q(x)}, \tag{7}$$

где $p(x)$, $q(x)$ – дискретные распределения вероятностей по X. При этом $D_{\text{KL}}(p,q)\neq D_{\text{KL}}(q,p)$ и $D_{KL}(p,q)\geq 0$.

Существуют различные подходы к симметризации расстояния КЛ. Для преодоления данного недостатка была выбрана процедура симметризации расстояния по схеме «среднего сопротивления» (Resistor Average), которая дает наименьшую среднюю ошибку при использовании оптимального байесовского классификатора [4]

$$\frac{1}{D_{\text{symKL}}(p,q)}=\frac{1}{D_{\text{KL}}(p,q)}+\frac{1}{D_{\text{KL}}(q,p)}.$$

Расстояние КЛ присутствует в альтернативном описании алгоритма ЕМ как расстояние между распределением вероятностей наблюдаемых переменных и условным распределением ненаблюдаемых параметров.

В приложении к алгоритму БУ, расстояние КЛ будет рассчитываться между моделями и с учётом наблюдаемых последовательностей. Формулы (8), (9) показывают алгоритм расчета расстояний между моделями λ_1, λ_2. С учётом того, что модель λ_1 обучалась на последовательности O_1, а модель λ_2 – на последовательности O_2 соответственно. Под выражениями $\text{seqLenght}(O_1)$, $\text{seqLenght}(O_2)$ понимаются длины соответствующих цепочек наблюдаемых переменных:

$$D_{\text{KL}}(\lambda_1, \lambda_2) = \frac{P(O_1|\lambda_1) - P(O_1|\lambda_2)}{\text{seqLenght}(O_1)}, \tag{8}$$

$$D_{\text{KL}}(\lambda_2, \lambda_1) = \frac{P(O_2|\lambda_2) - P(O_2|\lambda_1)}{\text{seqLenght}(O_2)}. \tag{9}$$

Таким образом, расстояние КЛ для СММ рассчитывается как отношение разности правдоподобий порождения наблюдаемой последовательности моделями к длине наблюдаемой последовательности.

Симметризованный вариант расстояния показан в выражении (10).

$$D_{\text{symKL}}(\lambda_1, \lambda_2) = \frac{1}{\frac{1}{D_{\text{KL}}(\lambda_1, \lambda_2)} + \frac{1}{D_{\text{KL}}(\lambda_2, \lambda_1)}} \tag{10}$$

В приложении к системе распознавания РПС, используя выражения (8), (9), (10), можно построить матрицу взаимных расстояний между N_{mk} СММ, восстановленными на первом этапе. Матрица M_D будет иметь вид, представленный в выражении (11).

Результатом данного этапа является симметричная матрица M_D с нулями на главной диагонали, заполненная значениями симметризованных расстояний КЛ, рассчитанных попарно между всеми моделями кластеров относительно последовательностей, на которых производилось обучение моделей.

	λ_1	λ_2	λ_3
λ_1	*0*	$D_{\text{symKL}}(\lambda_1, \lambda_2)$	$D_{\text{symKL}}(\lambda_1, \lambda_3)$
λ_2	$D_{\text{symKL}}(\lambda_1, \lambda_2)$	*0*	$D_{\text{symKL}}(\lambda_2, \lambda_3)$
λ_3	$D_{\text{symKL}}(\lambda_1, \lambda_3)$	$D_{\text{symKL}}(\lambda_2, \lambda_3)$	*0*

(11)

Третий этап – кластеризация и обучение моделей полученных кластеров.

На последнем этапе обучения производится выделение схожих кластеров по принципу минимизации межкластерного расстояния по матрице, полученной на втором этапе. Затем производится повторное обучение СММ по вновь образованным кластерам на цепочках машинных команд, объединённых в один кластер.

В качестве алгоритма кластеризации был выбран алгоритм восходящей иерархической классификации, в основе которого лежит обобщённая алгомеративная процедура [5]. Данный алгоритм является иерархическим, не зависит от выбора начальной точки, варьируемым параметром при этом является число кластеров N_{cl}.

На первом шаге каждый объект является отдельным кластером – результат первого этапа, N_{mk} СММ. На следующем шаге объединяются два ближайших объекта, которые образуют новый кластер, определяются расстояния от этого кластера до всех остальных моделей, и размерность матрицы расстояний M_{D} сокращается на единицу. Итерации повторяются до тех пор, пока не останется заданное N_{cl} число кластеров.

В приложении к системе распознавания данный алгоритм будет выглядеть следующим образом:

а) поиск по матрице расстояний M_{D} минимального значения межкластерного расстояния – выбор двух кластеров для объединения;

б) объединение кластеров, выбранных на шаге а);

в) пересчёт матрицы межкластерных расстояний.

На данном этапе возможно управление числом кластеров исходя из ограничений, указанных исследователем.

Ниже подробно рассмотрена процедура объединения на примере матрицы расстояний M_{D}, выражение (11). Пусть были выбраны модели λ_1, λ_3, обученные на последовательностях O_1, O_3 соответственно. Тогда модель λ'_1, являющаяся объединением, будет обучена на последовательности O_1O_3, представляющей собой конкатенацию двух последовательностей O_1, O_3, а выражения (8), (9) для расчёта расстояния КЛ примут вид:

$$D_{\mathrm{KL}}(\lambda'_1, \lambda_2) = \frac{P\left(O_1O_3 \middle| \lambda'_1\right) - P\left(O_1O_3 \middle| \lambda_2\right)}{\mathrm{seqLenght}(O_1O_3)},$$

$$D_{\mathrm{KL}}(\lambda_2, \lambda'_1) = \frac{P\left(O_2 \middle| \lambda_2\right) - P\left(O_2 \middle| \lambda'_1\right)}{\mathrm{seqLenght}(O_2)}.$$

На каждой итерации производится обучение новой модели на последовательности, представляющей собой объединение двух обучающих последовательностей выбранных кластеров, и пересчёт матрицы взаимных расстояний. Таким образом, после объединения размерность матрицы расстояний M_{D} сокращается на единицу.

Результатом данного этапа являются N_{cl} кластеров, обученных на последовательностях наблюдаемых значений, полученных из выборки незаражённых файлов в результате процедуры кластеризации.

Результатом процедуры обучения являются $2N_{\mathrm{cl}}$ кластеров, полученных из выборки незаражённых файлов и класса файлов РПС, которые представляют собой базу данных моделей кластеров, используемых в процессе распознавания.

Процедура распознавания включает в себя два основных этапа:

1) расчёт правдоподобия порождения цепочки каждым из кластеров по алгоритму прямого хода;

2) выбор кластера по критерию максимального правдоподобия.

На первом этапе на вход системы распознавания поступает выделенная из анализируемого файла ЦМК O_{a}. На основе данной последовательности наблюдений для каждой СММ из базы моделей кластеров по алгоритму прямого хода производится расчёт правдоподобия порождения данной цепочки $P\left(O_{\mathrm{a}} \middle| \lambda_i\right), i = \overline{1,2N_{\mathrm{cl}}}$.

На втором этапе по критерию максимального правдоподобия производится выбор кластера и принимается решение о принадлежности анализируемого файла к определённому классу, в зависимости от того, кластер какого класса был выбран

$$n_{\mathrm{ml}} = \max_{\lambda_1,\ldots,\lambda_{2N\mathrm{cl}}} P(O_{\mathrm{a}} | \lambda_i).$$

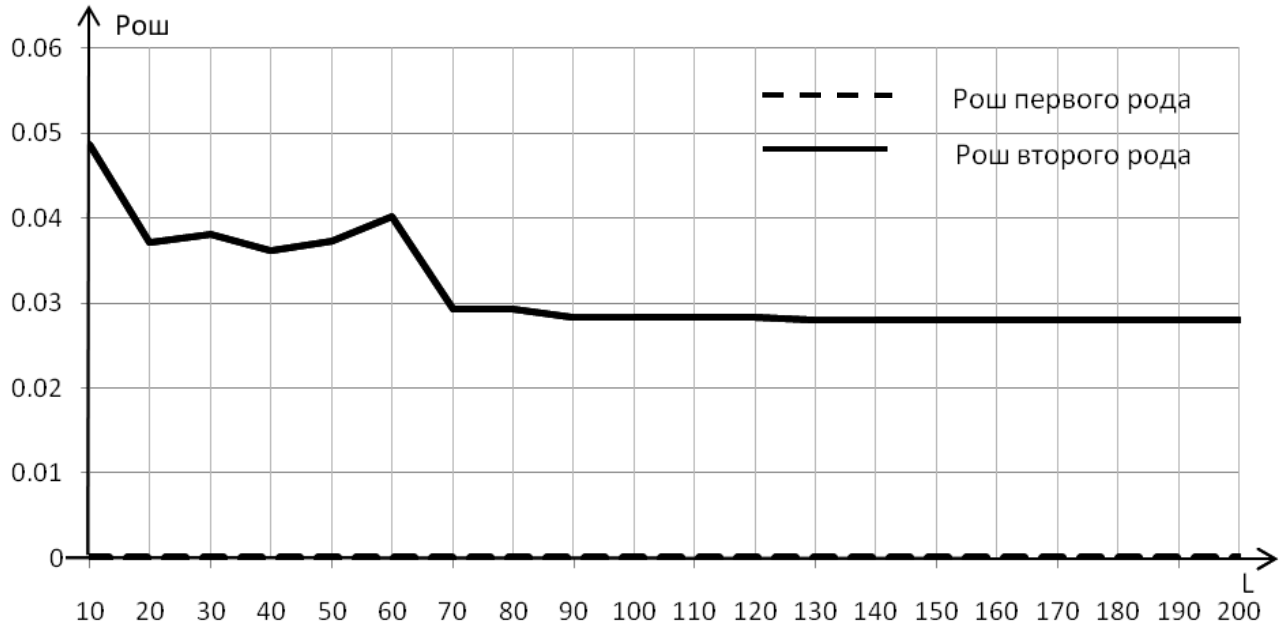

Рис.2 Зависимость ошибок первого и второго рода от длины анализируемой последовательности машинных команд при распознавании файловых вирусов на основе СММ

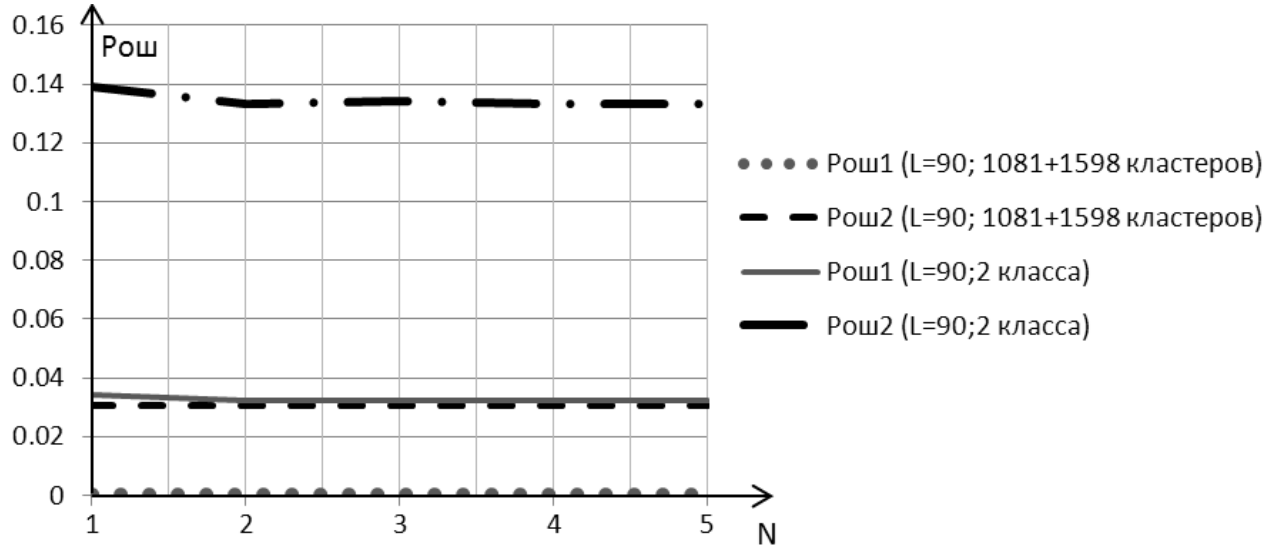

Рис.3 Зависимость ошибок первого и второго рода от количества скрытых состояний при распознавании файловых вирусов на основе СММ

Программная реализация механизма распознавания РПС на основе СММ позволила оценить эффективность применения данного подхода и показала существенное снижение значений оценок вероятностей ошибок первого и второго рода по сравнению со структурным стохастическим механизмом (рис. 2, 3), при условии выделения кластеров по количеству цепочек машинных команд из обучающей выборки каждого из классов машинного кода. При этом оценка вероятности ложных срабатываний составила порядка 0.003, оценка вероятности пропуска цели – 0.027, при длине анализируемой последовательности 90 машинных команд и 2 скрытых состояниях СММ.

На рис. 3 сплошная линия и штрихпунктирная отображают зависимость значений ошибок первого и второго рода от количества скрытых состояний при длине анализируемой последовательности в 90 машинных команд без применения процедуры внутриклассовой кла-

стеризации, каждый кластер при этом обучался на всей выборке цепочек машинных команд соответствующего класса машинного кода. Результаты приведены ниже:

$$\begin{cases} p_{ош1} = 0.032, \\ p_{ош2} = 0.133. \end{cases}$$

Две пунктирные линии рис. 3 отображают значения ошибок первого и второго рода для случая обучения кластеров по количеству цепочек машинных команд из обучающей выборки каждого из классов машинного кода (1081 – объём выборки файловых вирусов, 1598 – объём выборки незараженных файлов). Результаты приведены ниже:

$$\begin{cases} p_{ош1} = 0.003, \\ p_{ош2} = 0.027. \end{cases}$$

Из анализа графиков на рис. 2 можно сделать вывод, что минимально необходимой длиной анализируемой последовательности является цепочка в 90 машинных команд.

Приведённые результаты позволяют обосновать необходимость проведения процедуры внутриклассовой кластеризации машинного кода, а также показать зависимость значений ошибок первого и второго рода от таких параметров, как длина анализируемой последовательности и количество скрытых состояний модели.

В заключение необходимо отметить, что несмотря на значительное уменьшение ошибок первого рода в процессе распознавания, остаётся нерешённой задача выбора рационального числа кластеров. Решение данной задачи является направлением дальнейших исследований.

Литература

1. *Мацкевич А. Г., Козачок В. И.* Математическая модель системы стохастического структурного распознавания файловых вирусов // Безопасность информационных технологий. – М: Издательство МИФИ, 2007. – Выпуск 3. – С. 44–50.
2. *Козачок А. В., Мацкевич А. Г.* Модификация структурного метода распознавания вирусов // Информация и безопасность. Воронеж: Издательство «Воронежский государственный технический университет», 2010. – Выпуск 1, том 13. - С. 33–36
3. *Рабинер Л. Р.* Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: Обзор. // ТИЭР – М: Наука, 1989. – Выпуск 2. – Том 77. – С. 86-102.
4. *Кельберт М. Я., Сухов Ю. М.* Вероятность и статистика в примерах и задачах, том II: марковские цепи как отправная точка теории случайных процессов и их приложения. – М: Издательство МЦНМО, 2009. – 571 с.
5. *Don H. Jonson, Sinan Sinanovic* Symmetrizing the Kullback-Leibler distance. – IEEE Trans. on Comm. Tech., Volume 46, Number 1, p. 52-60, 2007.
6. *Мандель И. Д.* Кластерный анализ. – М: Издательство «Финансы и статистика», 1988. – 176 с.

Статья поступила в редакцию 21.12.2011;
переработанный вариант — 22.05.2012

Козачок Александр Васильевич
инженер Академии ФСО России (302034, Орёл, ул. Приборостроительная, 35) тел. (4862) 549725, e-mail: totrin@list.ru.

Mathematical Model for Destructive Software Tools Recognition Based on Markov's Models

A. Kozachok

This article observes destructive software tools recognition based on hidden Markov's models. Mathematical model of recognition system is considered and assessment results of it's effectiveness are presented.

Keywords: virus protection, destructive tools, hidden Markov's models, clustering.