ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ СТРУКТУРНОЙ ИДЕНТИФИКАЦИИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ КРИПТОСИСТЕМ

А.А. Горбунов, А.А. Разумов

Нижегородский госуниверситет

Существует целый ряд подходов к оценке стойкости криптосистем (КС), в том числе по расстоянию единственности, к которым относятся как классический подход К. Шеннона [1], так и подход, основанный на определении *оптимальных базовых параметров* (ОБП) входных и выходных текстовых процессов криптографических преобразователей (шифраторов, дешифраторов) КС [2]. Второй из указанных подходов базируется на получении динамических математических моделей (ММ) КС путем структурной идентификации, опирающейся только на имеющиеся в наличии открытые тексты и шифротексты.

Для решения задачи структурной идентификации динамических моделей КС, реализован эффективный алгоритм поиска оптимальных базовых параметров

$$\text{ОБП} = \{q, n\}$$

для экспериментально получаемых входных $\{u_0, \ldots, u_{M-1}\}$ и выходных $\{y_0, \ldots, y_{M-1}\}$ текстовых данных ($t = 0, \ldots, M-1$). При этом осуществляется минимизация энтропийной функции $E(q, n(q)) = n(q)\cdot\log q$ неавтономного источника текста. Границы области поиска базового параметра q (числа уровней квантования текста) задаются, исходя из априорных сведений об идентифицируемой модели объекта.

Реализация алгоритма вычисления базового параметра $n = n(q)$ обеспечивает по возможности наиболее быстродействующий вариант нахождения минимального значения порядка n прогнозирующего оператора с непротиворечивой таблицей истинности текстовой q-уровневой последовательности. Это означает, что, любым n идущим подряд символам $\{\ldots, u_{t-n+1}, \ldots, u_t, \ldots\}$ всегда должен соответствовать только единственный вариант последующего символа u_{t+1}, прогнозируемого по данной n-последовательности.

Можно выделить по крайней мере три подхода к построению алгоритмов для решения этой задачи в зависимости от организации доступа к наборам анализируемых n-последовательностей [3, 4].

1. Алгоритмы, основанные на непосредственном переборе всех n-последовательностей в тексте с оценкой времени работы $O(M^2)$ и ограниченной областью практической применимости для случаев сравнительно коротких текстовых последовательностей.

2. Бинарный поиск среди n-последовательностей, основанный на их индексировании в соответствии с некоторым правилом упорядочивания. Общее время работы таких алгоритмов можно оценить уже как $O(M\log M)$, что дает

возможности производить при помощи них обработку значительно более длинных последовательностей. Однако при данном подходе имеются дополнительные затраты памяти для поддержки упорядоченности n-последовательностей.

3. Построение суффиксного дерева по обрабатываемой текстовой последовательности (например, при помощи алгоритма Укконена [5]) с линейной оценкой $O(M)$ времени работы алгоритма, что с теоретической точки зрения является неким нижним пределом быстродействия для всех алгоритмов, обращающихся к каждому символу обрабатываемого текста. Тем не менее, практическая реализация построения суффиксного дерева требует существенных затрат памяти как при достаточно большой длине M обрабатываемой текстовой последовательности, так и при большой размерности q её алфавита.

Вышеуказанные алгоритмы поиска совпадающих n-последовательностей были программно реализованы на языке C# в среде разработки программного обеспечения Microsoft Visual Studio 2010. Программная реализация алгоритма нахождения ОБП и соответствующих им параметров стойкости была протестирована на экспериментальных текстовых данных, считанных с входа и выхода шифраторов следующих криптосистем: DES в режимах электронной цифровой книги (ECB), сцепления блоков (CBC), обратной связи по выходу (OFB); потоковой криптосистемы RC4; ряда шифрующих линейных цифровых автоматов (для шифров Цезаря, Вижинера и пр.) Компьютерный эксперимент осуществлялся на аппаратной платформе со следующей конфигурацией: Intel Core 2 Duo CPU E8400 3GHz, RAM 4 Gb, ОС Windows Vista Ultimate SP2 x86. В результате проведения компьютерного эксперимента на основе перечисленных выше алгоритмов было получено практическое подтверждение приведенных зависимостей времени работы от длины обрабатываемых текстовых последовательностей.

[1] Шеннон К. Работы по теории информации и кибернетике./Пер. Писаренко В.Ф.– М.: Иностранная литература, 1963.
[2] Горбунов А.А., Кирьянов К.Г. //Вестник Нижегородского университета им. Н.И. Лобачевского. Серия «Радиофизика». 2005. Вып. 1(3). С.185.
[3] Горбунов А.А. //В кн.: Тр. XIII-й научн. конф. по радиофизике. 7 мая 2009 г. /Ред. С.М. Грач, А.В.Якимов. – Н.Новгород: Изд-во «ТАЛАМ», 2009. С.225.
[4] Горбунов А.А. //Доклады Томского государственного университета систем управления и радиоэлектроники. 2009. №1 (19), ч.2. С. 21.
[5] Гасфилд Д. Строки, деревья и последовательности в алгоритмах. – СПб.: Невский диалект; БХВ-Петербург, 2003, 654 с.

ОЦЕНКА ПОГРЕШНОСТИ СИНХРОНИЗАЦИИ ШКАЛЫ ВРЕМЕНИ ПРИЕМНИКА-КОМПАРАТОРА СИГНАЛОВ ГЛОНАСС/GPS С ЭТАЛОННОЙ ШКАЛОЙ

В.В. Акулов

ФГУП ННИПИ «Кварц»

Целью данной работы является вывод и анализ выражений для оценки погрешности синхронизации шкалы времени (ШВ) системы синхронизации (СС) приемника-компаратора сигналов СРНС ГЛОНАСС/GPS по алгоритму, учитывающему тренд частоты его опорного генератора и рекурсивную фильтрацию данных о расхождении шкал, а также разработка рекомендаций по выбору оптимальных параметров алгоритма.

Ранее были рассмотрены различные алгоритмы работы СС приемника-компаратора [1], которые не учитывают систематическое изменение частоты внутреннего рубидиевого генератора, что приводит к систематическому изменению (тренду) измеряемой в приборе величины расхождения ШВ приемного устройства сигналов СРНС и идеальной ШВ *x*.

В приемнике-компараторе ЧК7-56 формирование ШВ осуществляется с учетом тренда частоты внутреннего опорного генератора. Исходными данными для очередной коррекции частоты и ШВ приемника-компаратора служат параметры тренда и время $T_{изм}$ между коррекциями. Среднее систематическое относительное изменение частоты ν за $T_{изм.}$ определяется методом наименьших квадратов за N с. Пусть $T_i = 1, 2, \ldots, N$ – номера моментов времени $t_i = 1, 2, \ldots, N$ *c*, в которые производится запись данных x_i (через N *c* происходит коррекция частоты внутреннего генератора и ШВ приемника-компаратора). Пренебрегая постоянным членом c_o в уравнении тренда при большом N (при малой начальной отстройке внутреннего генератора), предполагая, что вычисляемые процессором величины относительных отстроек частоты за 1с являются взаимно независимыми, получим выражение для дисперсии $\sigma^2_{f_{тр}}$ величины $\delta f_{тр}$, равной относительному изменению интервала времени между «истинной» ШВ и ШВ прибора при $T_{изм} = const$.

Далее, предполагая взаимную независимость случайной величины $\delta f_{тр}$, которая зависит также от внутренних параметров опорного генератора (старение, износ активных элементов и др.) и определяется процессором по окончании времени измерения $T_{изм}$ и случайной составляющей $\sigma_{дискр.}$ величины погрешности коррекции ШВ из-за дискретности временного счета формирователя шкал времени (ФШВ) $\Delta t_{дискр}$, можно записать общее выражение для погрешности синхронизации в приемнике-компараторе ШВ опорного генератора с учетом его тренда, определяемой через среднее квадратическое отклонение относительного изменения интервала времени между «истинной» ШВ (ШВ UTC) и ШВ прибора:

$$\sigma_{ШВ_{тр}} = \sqrt{\sigma^2_{f_{тр}} + \sigma^2_{дискр.}} = \sqrt{\cdot \frac{36\sigma^2_{СРНС}}{N \cdot \Delta t^2} + \left(\frac{\left|\Delta t_{дискр.}\right|}{T_{изм.}} \right)^2} \; .$$

Из данного выражения следует, что $\sigma_{ШВ_{тр}}$ тем меньше, чем больше $T_{изм} = N \cdot \Delta t$, которое, в свою очередь, тем больше, чем меньше величина тренда, чтобы могла произойти коррекция ШВ (сдвиг по частоте генератора и ШВ по времени). На практике выдаваемая ЧК7-56 ШВ после обработки имеет достаточно большие сдвиги, обусловленные, прежде всего как сдвигами ШВ с приемоизмерительного модуля за счет, например, смены видимой группировки НКА, так и сдвигами самой системной ШВ СРНС. С целью их сглаживания осуществлялась фильтрация экспериментальных данных по рекурсивному выражению: $y_n = k \cdot y_{n-1} + (1 - k) \cdot x_n$, где y_n – усредненное значение текущих данных о расхождении эталонной ШВ и ШВ приемника-компаратора; x_n – текущее значение данных о расхождении эталонной ШВ и ШВ приемника-компаратора; k – параметр рекурсивного фильтра, $0 \le k < 1$. Известно [2], что такой фильтр эквивалентен RC-фильтру с коэффициентом передачи $k = e^{-\frac{T}{\tau}}$, где T – период дискретизации, с; τ – постоянная фильтра, с. Из последнего выражения следует $\tau = -\frac{T}{\ln k}$. Предполагая, что каждое значение y_n при большом значении n и малых значениях k является почти независимым от предыдущего значения y_{n-1}, а в большей степени зависит от значения x_n, которое, в свою очередь, является независимым от значения x_{n-1}, что определяется взаимной независимостью положений метки времени с приемно-измерительного модуля сигналов СРНС в каждую секунду, и применяя оператор вычисления дисперсии к выражению для рекурсивной фильтрации получим: $\sigma^2_{y_n} + k^2 \cdot \sigma^2_{y_{n-1}} = (1-k)^2 \cdot \sigma^2_{x_n}$, где $\sigma_{y_n} = СКО_{y_n}$, $\sigma_{x_n} = СКО_{x_n}$. Считая далее, что при больших значениях n $\sigma_{y_n} = \sigma_{y_{n-1}} = \sigma_y$, а $\sigma_{x_n} = \sigma_x$, из последнего выражения следует: $\sigma^2_y = \frac{(1-k)^2}{1+k^2} \cdot \sigma^2$.

Очевидно, что полученным выражением можно пользоваться только при малых значениях k, так как при больших k существенное значении имеет взаимозависимость случайных величин в формуле для рекурсивной фильтрации, что приводит к необходимости вычисления ковариаций указанных величин. Последнее на практике не всегда возможно, так как требует знания совместных распределений вероятностей. После рекурсивной обработки экспериментальных данных, изменяя параметр k по минимуму СКО обработанных данных, можно выбрать его наиболее приемлемое значение, а из выражений для RC-фильтра–оптимальное соотношение T и τ.

[1] Акулов В.В., Кирьянов К.Г.Система синхронизации шкал времени по сигналам ГЛОНАСС/GPS //В кн.: Тр. XIII-й научн. конф. по радиофизике. 7 мая 2009 г. /Ред. С.М. Грач, А.В.Якимов. – Н.Новгород: Изд-во «ТАЛАМ», 2009. С.223.

[2] Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. Вып.1. – М.: Мир, 1974.

ПРОГНОЗИРОВАНИЕ ПОВЕДЕНИЯ ПРОЦЕССОВ НА ОСНОВЕ ИХ ОПТИМАЛЬНЫХ БАЗОВЫХ ПАРАМЕТРОВ

Е.С. Кузнецов

Нижегородский государственный технический университет

В данной работе рассматривается метод прогнозирования, основанный на оптимальных базовых параметрах (БП) [1]: q – число уровней квантования выборок процесса (1), Δt – шаг квантования по времени выборок данных (1)) и n – число аргументов так называемого «прогнозирующего оператора» (ПО) по критерию минимума энтропии.

Требуемая для процедуры прогнозирования структурная идентификация исходных векторных временных рядов, используемая для синтеза ПО, осуществляется фактически одновременно при нахождении оптимальных базовых параметров (БП) временных рядов длиной M:

$$\{y_i^v\}^T; i \in [0, M-1]; v \in [1, r], \quad (1)$$

где r – количество компонент векторного процесса.

Определения оптимальных базовых параметров заключается в нахождении такой тройки $\Delta t, q, n$ при которой у последовательности (1) будет минимальная энтропия. Причём $T/M_{max} \le \Delta t \le T/M_{min}$ (где T – длительность исходного процесса), а $q_{min} \le q \le q_{max}$. Порядок прогнозирующего оператора определяется как минимальное n, при котором по одной и той же n-последовательности отсчетов $(y^1{}_{k-n+1}, y^1{}_{k-n+2}, \ldots, y^1{}_k, \ldots, y^r{}_{k-n+1}, y^r{}_{k-n+2}, \ldots, y^r{}_k)$ прогнозируются одинаковые значения $(y^1{}_{k+1}, \ldots, y^r{}_{k+1}) = f_k$.

Затем по последовательности (1) строится ПО для любого $k = n, n+1, \ldots, M-1$ в виде q-значной нелинейной логической функции с оптимальными аргументами, что является существенным отличием от известных методов прогнозирования векторных процессов

$$(y^1{}_{k+1}, \ldots, y^r{}_{k+1})^T = f((y^1{}_{k-n+1}, y^2{}_{k-n+1}, \ldots, y^r{}_{k-n+1})^T, \ldots, (y^1{}_k, y^2{}_k, \ldots, y^r{}_k)^T) \quad (2)$$

или эквивалентной таблицы истинности (ТИ). Строки ТИ ПО строятся по всем идущим подряд n отсчетам и следующего за ними отсчета, в качестве прогнозируемого ими символа.

Построение ПО для $k \ge M$ заключается в пошаговом построении продолжения ТИ с $(M-n+1)$-й по $(M+sf)$-ю строку, где $sf = 1, \ldots, Lf$, а Lf – номер «прогнозного горизонта» для пополнения выборок данных (2), имеющихся в исходной ТИ. Для определения $(y^1_{M-1+sf}, \ldots, y^r_{M-1+sf})^T$ используется последовательное сравнение $(y^1_{M-2+sf}, \ldots, y^r_{M-2+sf})^T$-ой n-последовательности $((y^1_{M-1-n+sf}, \ldots y^1_{M-2+sf})^T, \ldots, (y^r_{M-1+n+sf} \ldots, y^r_{M-2+sf})^T)$ со всеми n- последовательностями, уже имеющимися в исходной таблице, рассматриваемыми как опорные («эталонные») по критерию «минимума расстояния» между n- последовательностями (вариант «а») [1].

Возможна модификация формулы (2) с учетом влияния классов эквивалентности на критерий «минимума расстояния» (вариант «б») [2].

При определении числа возможных продолжений q-уровневой последовательности на заданную длину было найдено соотношение для наиболее вероятного варианта продолжения (вариант «в») [3].

Результаты сравнения методов прогнозирования годового прироста ширины колец деревьев сведены в таблицу. При этом оценка точности прогноза проводилась по формуле:

$$\delta(s) = (y(s) - y'(s))/(y'_{\max} - y'_{\min}), \quad (3)$$

где y' – исходный (реальный) процесс, s – шаг прогноза, $y'_{\max}=\max\{y'_i\}$, $y'_{\max}= \min\{y'_i\}$, $i\in [0, M–1]$. Сравниваются исходный процесс (y' (s), $s\in[M–L, M–1]$) и спрогнозированный процесс ($y(s)$, $s\in[M–L, M–1]$) на основе исходного процесса с отброшенным концом (y' (s), $s\in[0, M–L–1]$), где L – длина прогноза.

Табл.

s	1	2	3	4	5
$\delta(s)_а$	0	-0,04	-0,04	0,04	-0,25
$\delta(s)_б$	0,05	0,05	-0,35	-0,4	-0,8
$\delta(s)_в$	0	0,2	-0,05	0,15	-0,45

Исследования, проведенные с помощью разработанной информационной системы прогнозирования, показали, что точность прогноза, получаемая на выборке данных природного процесса сопоставима с результатами известных программ («Эвриста», «Statistica»).

[1] Кирьянов К.Г., Кузнецов Е.С. //В кн.: Тр. XIV-й научн. конф. по радиофизике. 7 мая 2010 г. /Ред. С.М. Грач, А.В.Якимов. – Н.Новгород: Изд-во ННГУ, 2010. С.154.

[2] Кирьянов К.Г., Кузнецов Е.С. //В кн.: Тр. XII-й научн. конф. по радиофизике. 7 мая 2008 г. /Ред. С.М. Грач, А.В.Якимов. – Н.Новгород: Изд-во «ТАЛАМ», 2008. С.271.

[3] Кирьянов К.Г., Кузнецов Е.С. //В кн.: Тр. XIII-й научн. конф. по радиофизике. 7 мая 2009 г. /Ред. С.М. Грач, А.В.Якимов. – Н.Новгород: Изд-во «ТАЛАМ», 2009. С.214.

МЕТОД ОЦЕНКИ КРИПТОСТОЙКОСТИ ПАРОЛЕЙ УЧЕТНЫХ ЗАПИСЕЙ ПОЛЬЗОВАТЕЛЕЙ

А.А. Новокрещенов

Нижегородский государственный технический университет

Процедура обеспечения информационной безопасности вычислительной системы представляет собой комплекс мер по сетевой безопасности, антивирусной защите, контролю прав пользователей, надежной аутентификации и т.д. Каждый из этих аспектов является важным, и его недооценка может привести к серьезным негативным последствиям.

В данной работе рассматривается проблема ненадежных учетных записей и предлагается одно из возможных решений данной проблемы. Проблема ненадежных учетных записей является следствием выбора пользователями «плохих» паролей, которые могут быть скомпрометированы злоумышленником. Как результат, получение злоумышленником полномочий пользователя-владельца учетной записи в системе.

Одним из способов защиты от «плохих» паролей является оценка их криптостойкости самим администратором, обеспечивающим безопасность вычислительной системы. В случае обнаружения таких паролей, администратор в праве требовать от пользователя смены пароля. Все что нужно администратору, для выполнения данной задачи – это хэши паролей.

В данной работе предлагается способ восстановления пароля по его хэшу. В общем виде решение такой задачи является крайне сложной, а в большинстве случаев – невыполнимой задачей. Задача восстановления пароля может стать практически реализуемой благодаря следующим допущениям:

– администратор знает алгоритм, который используется системой аутентификации для шифрования паролей;
– пароль – короткая последовательность длиной не более десяти символов;
– алфавит пароля – это множество, включающее только те символы, которые можно ввести с клавиатуры, т.е. набор ASCII без управляющих символов, символов псевдографики и букв русского алфавита;
– в силу того, что пароль генерируется человеком, он может содержать осмысленные слова.

Предположение об алфавите пароля правомерно по причине того, что подавляющее большинство пользователей по привычке не использует кириллические символы при создании паролей. Алфавит пароля представляет собой следующее множество: **A**={A÷Z, a÷z, 0÷9, !, @, #, $, %, ^, &, *, (,), ;, [,], {, }, +, /, <, >, ., ?, |, `, ~, ‘, =, –}, его мощность составляет M=96.

С учетом данных допущений представляется целесообразным проведение сразу нескольких типов анализа: анализа хэша по словарю, инкрементного анализа его модификации и полного перебора. Словарный и инкрементный типы анализа являются эффективными для пользовательских паролей в силу последнего допущения.

Для выполнения указанных типов анализа предлагается использовать гетерогенную схему вычислений с использованием графического процессора (GPU) [1]. Анализ методом полного перебора представляет собой хорошо распараллеливаемую задачу в силу того, что вычисление отдельной перестановки, расчет хэша и сверка с исходным хешем не зависит от других перестановок. Для ее выполнения предлагается использовать вычислительные ресурсы графического процессора. Современные графические процессоры – это многоядерные процессоры с массово параллельной архитектурой, которые как нельзя лучше подходят для выполнения хорошо распараллеливаемых задач [1].

Объем вычислений при выполнении словарного и инкрементного анализа меньше по сравнению с полным перебором, поэтому данные задачи предлагается выполнять на центральном процессоре.

Общая схема вычислений приводится на рисунке.

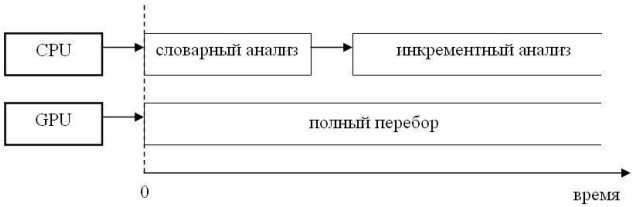

Рис.

В рамках данной работы было реализовано программное обеспечение для выполнения указанных вычислений. В таблице приводятся результаты работы разработанного программного обеспечения относительно нескольких паролей. Пароли восстанавливались из хэшей, зашифрованных алгоритмом MD5.

Табл.

Пароль	время работы на CPU	время работы на GPU
hello123	1 с	≈576 дн.
acf^#	7 ч.	50 с
acf8^#	≈29 д.	2,5 ч.

Время восстановления первого пароля на GPU и третьего на CPU является расчетным, оно получено на основании результатов восстановления паролей меньшей длины (3, 4, 5 символов).

Временные замеры показывают, что скорость перебора паролей с использованием GPU составляет $\approx 13{,}5 \times 10^7$ паролей в секунду.

В качестве CPU в данной работе использовался процессор Intel Pentium Dual Core (2,6 GHz), в качестве GPU – NVIDIA GT9600 (64 скалярных процессора, 500 MHz каждый).

[1] NVIDIA CUDA C Programming Guide version 3.2. 2011. P. 11.

ПРИМЕНЕНИЕ АЛГОРИТМА ВЫЧИСЛЕНИЯ ЛОКАЛЬНОГО СУФФИКСНОГО ВЫРАВНИВАНИЯ В ЗАДАЧАХ ИССЛЕДОВАНИЯ ИСПОЛНЯЕМОГО КОДА

А.Н. Дмитришин, А.В. Корюкалов, Л.Ю. Ротков

Нижегородский госуниверситет

В настоящее время существует актуальная задача автоматизации идентификации реализаций различных алгоритмов в исполняемых кодах программ, удовлетворяющих определенным условиям. Такие задачи могут встречаться при детектировании вредоносного программного обеспечения, а также при сертификации программного обеспечения на соответствие требованиям по безопасности информации.

Предложенный подход к автоматизированному исследованию исполняемого кода с использованием алгоритмов анализа строк [1] предполагает поиск в строке, соответствующей исследуемому коду, подстрок, обладающих высоким сходством со строками-шаблонами. Каждая такая строка-шаблон является представлением различных реализаций одного алгоритма (или представлением различных кодов, обладающих определенным общим свойством). Пополнять библиотеку таких шаблонов предлагается следующим образом.

С использованием полученного опытным путем оптимального преобразования графа в строку f_{opt}: $G \to S$ [1] вычисляются строки S_1 и S_2, соответствующие двум различным реализациям одного алгоритма. Такие реализации могут представлять собой компилированный с использованием различных компиляторов (или различных параметров компиляции) исходный код или разные версии одной вредоносной программы.

В строках S_1 и S_2 ищутся пары подстрок, обладающие определенным сходством (при этом минимально допустимая степень сходства определяется количественно).

Для поиска пар похожих подстрок используется решение задачи локального выравнивания [2], позволяющее найти в двух строках подстроки, обладающие наибольшим сходством.

При решении задачи локального выравнивания используется алгоритм локального суффиксного выравнивания строк [2]. Исполнение данного алгоритма позволяет не только найти подстроки, обладающие максимальным сходством. При обратной трассировке имеется возможность определить все схожие подстроки с количественной характеристикой их сходства.

Найденные похожие подстроки являются претендентами на роль строк-шаблонов в библиотеку, используемую при автоматизированном исследовании.

Необходимо также отметить, что для реализации подхода [1] должно быть опытным путем определено не только оптимальное преобразование f_{opt}: $G \to S$, но и схема оценки локального выравнивания, влияющая на результат вычисления степени сходства: для ее вычисления алгоритм предполагает назначение весов для совпадения символов, несовпадения символов и вставки пробела.

[1] Дмитришин А.Н., Корюкалов А.В., Ротков Л.Ю. // В кн.: Тр. XIV научн. конф. по радиофизике. 7 мая 2010 г. /Под ред. С.М. Грача, А.В. Якимова. Н.Новгород: ННГУ, 2010. С. 274.

[2] Гасфилд Д. Строки, деревья и последовательности в алгоритмах. Информатика и вычислительная биология. – СПб: «БХВ – Петербург», 2000.

ОБНАРУЖЕНИЕ ТЕКСТОВОГО СПАМА МЕТОДОМ ГЕНЕТИЧЕСКИХ КАРТ

С.В. Корелов

Нижегородский госуниверситет

Рассматривается эффективность метода генетических карт текстов для обнаружения текстового спама [1, 2]. Исследовались генетические карты спам-текстов и

легальных рассылок. По рассчитанным генетическим картам принималось решение о принадлежности письма к спаму.

Представим работу антиспам-системы на основе метода генетических карт в виде схемы, изображенной на рисунке.

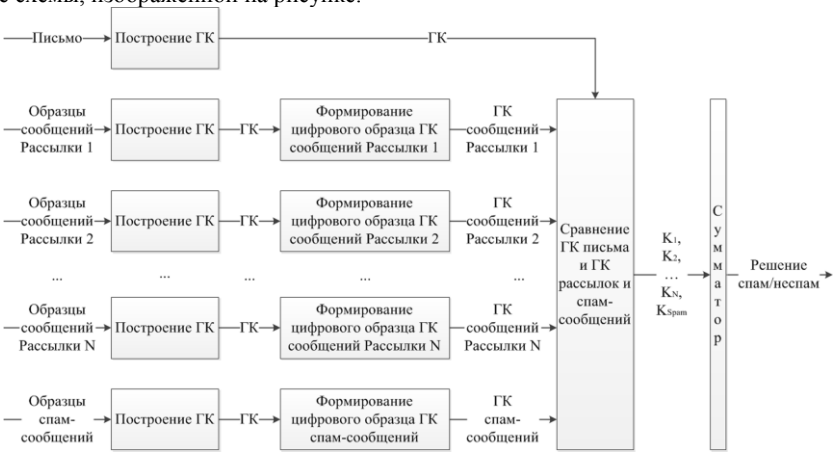

Рис.

Исходные текстовые письма (источники машинного текста) преобразованы в числовой вектор $y = (y_0, y_1, \ldots, y_{n-1}, y_n, \ldots, y_{M-2}, y_{M-1})$, где M – количество символов в письме. Критерием оценки применимости данного метода для выявления спама являются вероятности ошибок пропуска спама и принятие легальных писем за спам.

Способ обнаружения спама, основанный на построении генетических карт текстовых писем, реализован в виде компьютерной программы. При проведении эксперимента определены следующие значения параметров алгоритма построения генетических карт текстов: $q = 256$ (соответствует количеству символов в кодировке Windows-1251); $n = 2 \ldots 20$.

В эксперименте в качестве анализируемых текстов использованы 4 группы писем:

- 518 писем информационной рассылки портала www.securitylab.ru, полученных за период с 28.04.2009 по 04.03.2011;
- 374 письма информационной рассылки сайта www.security.nnov.ru, полученных за период с 27.04.2009 по 04.03.2011;
- 350 писем информационной рассылки сайта Xakep.RU, полученных за период с 28.04.2009 по 04.03.2011;
- 1881 спам-письмо, поступившее на почтовый домен grw.ru за период с 14:48 до 15:54 02.03.2011.

Все письма были ранжированы в порядке временного поступления.

На основе указанных выборок созданы базы генов каждой группы писем и рассчитаны коэффициенты принадлежности каждого письма к соответствующей группе. При этом коэффициент принадлежности к своей группе рассчитывался при участии генов только предыдущих по времени писем, а коэффициент принадлежно-

сти к другим группам при участии всех генов этих групп. Таким образом, в эксперименте были смоделированы наихудшие условия (максимальное количество чужих генов при минимальном количестве своих). Решение о принадлежности письма к той или иной группе принималось на основе максимального коэффициента. При этом письмо относилось к группе спам-писем только в случае строго большего неравенства соответствующего коэффициента. Таким образом, в ходе эксперимента искусственно ухудшены условия распознавания спама.

В результате проверки программой указанных групп писем получены следующие результаты.

Табл. 1

Значения n	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Кол-во писем, отнесенных к спаму, %	98,72	98,88	98,56	97,77	96,33	95,69	94,95	95,11	95,27

Табл. 1 (продолжение)

Значения n	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Кол-во писем, отнесенных к спаму, %	95,22	94,21	94,15	94,05	93,51	93,04	92,56	92,24	91,92	91,55

Таким образом, блок контент-анализа, функционирующий на основе метода генетических карт, позволил выявить от 91% до 98% спам-писем, пришедших на почтовый домен. Существует вероятность ухудшения результатов при применении данного метода на спам-письмах, приходящих на конкретный почтовый ящик.

[1] Кирьянов К.Г. // В кн.: Тр. VI научн. конф. по радиофизике. 7 мая 2002 г. /Под ред. А.В. Якимова. Н.Новгород: ТАЛАМ, 2002. С. 119.
[2] Кирьянов К.Г. // В кн.: Тр. III Междунар. конф. «Идентификация систем и задачи управления SICPRO'04». – М.: ИПУ РАН, 2004. С. 187.

ПРИМЕНЕНИЕ МЕТОДОВ DATA MINING ДЛЯ ФИЛЬТРАЦИИ СПАМА

О.И. Шкалябин

Нижегородский госуниверситет

В статье рассматриваются теоретические подходы к построению алгоритма фильтрации спама методами интеллектуального анализа данных (*Data Mining*).

Задачу фильтрации спама можно рассматривать как задачу классификации, в которой определяется принадлежность объектов к определенным классам на основании анализа совокупности признаков, характеризующих данный объект. Формально, если D – множество классифицируемых объектов, C – множество классов, и существует целевая функция $\Phi: C \times D \rightarrow \{0, 1\}$, значения которой известны на некотором конечном подмножестве объектов

D_L, то задача классификации – это построение функции Ф', максимально близкой к Ф. Функция Ф' называется классификатором, а подмножество D_L называется обучающей выборкой. Классификатор должен одинаково хорошо работать как на обучающей выборке, так и на всем наборе документов. В задаче фильтрации спама множество объектов составляют письма, которые необходимо классифицировать по двум классам: «спам» и «не спам». Как и для любой задачи классификации, необходимо решить основные проблемы: представление данных, формирование и сокращение размерности признакового пространства.

Предлагается взять за основу векторную модель представления данных [1], согласно которой определяется множество признаков T, характеризующих каждый объект d из D_L. Тогда объект d_j представляется в виде вектора, координатами которого являются весовые коэффициенты признаков: $\boldsymbol{d_j} = (w_{1j},\ldots,w_{|T|j})$. Множество всех допустимых значений признаков для всех объектов набора называется признаковым пространством.

Признаки разделяются на нетекстовые и текстовые. К первой группе относятся признаки, характеризующие оформление и статистические параметры электронного письма: размер письма, наличие, тип и размер вложений, кодировка и форматирование, различная информация из заголовка. Вторая группа признаков формируется из текстовой части письма, которая, по сути, является обычным текстовым документом. Чтобы сформировать признаковое пространство текстового документа, он должен рассматриваться как набор слов. При этом целесообразно не только учитывать все слова, входящие в документ, но и семантические связи между ними. Основным статистическим признаком текстовой части письма является наличие в нем различных слов. Метод определения весовых коэффициентов, основанный на статистике появлений слов в документе, когда $w_{ij} = f_{ij}$ – количество вхождений слова в документ, неэффективен. Предлагается использовать метод, учитывающий частоту появления слова во всей коллекции документов и различную длину слов, используя нормализацию

$$w_{ij} = \frac{f_{ij} \cdot \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{k=1}^{|d_j|}\left[f_{kj} \cdot \log\left(\frac{N}{n_k}\right)\right]^2}},$$

где N – общее количество документов в наборе, а n_i – количество документов в наборе, в которые хотя бы раз входит данное слово. Веса, вычисленные по данному методу, монотонно возрастают с ростом числа вхождений соответствующего признака в документ. Весовые коэффициенты нормируются по простому правилу:

$$\tilde{w}_{ij} = \frac{w_{ij}}{\sqrt{\sum_{s=1}^{|T|}\left(w_{sj}\right)^2}} \text{ так, чтобы } 0 \le w_{ij} \le 1.$$

Задача сокращения размерности данных решается различными методами, которые можно разделить на две группы: выбор наиболее информативных признаков и синтез признаков. Последовательное применение таких методов позволяет сократить признаковое пространство на два порядка, практически не потеряв при этом точности классификации [2].

В настоящее время применение методов *Data Mining* широко распространено, но не один не дает стопроцентного результата. Наиболее распространенные методы: наивный байесовский классификатор, линейный дискриминант Фишера, машина опорных векторов, нейронные сети. Исследования показывают преимущества того или иного метода в зависимости от тренировочного набора. Оценка по таким характеристикам, как время обучения, время и точность классификации, размер модели, дообучение и переобучение модели, показывает, что наиболее эффективным методом из перечисленных выше является SVM (машина опорных векторов) [2].

Перспективным направлением работы представляется построение композиций алгоритмов, позволяющих использовать лучшие характеристики отдельных классификаторов.

[1] Воронцов К.В. Машинное обучение. (http://www.machinelearning.ru)
[2] Розинкин А.Н. Система защиты от массовых несанкционированных рассылок электронной почты на основе методов Data Mining / Дисс. на соискание степени канд. физ.-мат. наук: 05.13.11– М.: МГУ, 2006, 110 с.