УДК 675.03.031.81 : 577.15

Профессор С.В. Жернаков, аспирант Г.Н. Гаврилов
(Уфимский авиац. гос. тех. ун-т) кафедра электроники и биометрических технологий.
тел. 89373571952
E-mail: grigorijgavrilov@mail.ru

Professor S.V Zhernakov, graduate G.N Gavrilov
(Ufa state aviation technical university) Department of electronics and biometric technology.
phone 89373571952
E-mail: grigorijgavrilov@mail.ru

Детектирование вредоносного программного обеспечения с применением классических и нейросетевых методов классификации

Detection of malicious software using classical and neural network classification methods

Реферат. Постановка задачи: спектр задач, решаемых современными мобильными системами типа Android, постоянно растет. Это объясняется, с одной стороны, теми потенциальными возможностями, которые реализуются на аппаратном уровне, а также их интеграцией с современными информационными технологиями, которые в свою очередь гармонично дополняют и создают мощные аппаратно-программные информационные комплексы, способные выполнять множество функций: в том числе и защиты информации. Увеличение информационных потоков, сложность обрабатываемых процессов и самой аппаратно-программной составляющей таких устройств как Android, заставляют разработчиков создавать новые средства зашиты, эффективно и качественно осуществляющие данный процесс. Это особенно актуально при разработке автоматизированных инструментальных систем, осуществляющих классификацию (кластеризацию) существующего программного обеспечения на два класса: безопасное и вредоносное программное обеспечение. Целью работы является повышение достоверности и качества распознавания современных встроенных средств защиты информации, а также обоснование и выбор методов, осуществляющих эти функции. Используемые методы: для реализации поставленной цели в работе анализируются и используются классические методы классификации, нейросетевые методы на основе стандартных архитектур, а также машина опорных векторов (SVM – машина). Новизна: в работе предложена концепция к использованию метода опорных векторов при идентификации вредоносного программного обеспечения, разработано методологическое, алгоритмическое и программное обеспечение, реализующее данную концепцию применительно к средствам мобильной связи. Результат: получены качественные и количественные характеристики программных средств защиты. Практическая значимость: предложена методика разработки перспективных систем защиты информации в мобильных средах типа Android. Представлен один из подходов к описанию поведенческого характера вредоносного программного обеспечения (на основе следующих действий вируса: отсутствует – просыпается – анализ слабых мест – действие: здоровый режим или атаки (угрозы)).

Summary. Formulation of the problem: the spectrum of problems solved by modern mobile systems such as Android is constantly growing. This is because on the one hand by the potential opportunities that are implemented in hardware, as well as their integration with modern information technologies, which in turn harmoniously complement and create powerful hardware and software information systems, capable of performing many functions, including pro- information boards. Increasing the flow of information, complexity of the processes and of the hardware and software component devices such as Android, forcing developers to create new means of protection, efficiency and qualitative performing the process. This is especially important in the development of automated systems instrumental performing classification (clustering) of existing software into two classes: safe and malicious software. The aim is to increase the reliability and quality of recognition of modern built-in security of information, as well as the rationale and the selection methods of carrying out these functions. The methods used are: to accomplish the goals are analyzed and used classical methods of classification, neural network method based on standard architectures, and support vector machine (SVM - machine). Novelty: The paper presents the concept of the use of support vector in identifying deleterious software developed methodological, algorithmic and software that implements this concept in relation to the means of mobile communication. Result: The obtained qualitative and quantitative characteristics-security software. Practical value: the technique of development of advanced information security systems in mobile environments such as Android. It presents an approach to the description of behavioral malware (based on the following virus: none - wakes - Analysis of weaknesses - the action: a healthy regime or attack (threat)).

Ключевые слова: Android, кластеризация, мобильные системы, классификация, программное обеспечение, обучающая выборка, иерархический метод, машина опорных векторов, нейронные сети.

Ключевые слова: Android, clustering, mobile systems, classification, software, training sample, a hierarchical method, support vector machine, neural networks.

Актуальность

За короткий период времени, система Android стала самой популярной мобильной платформой в мире. Изначально разработанная для смартфонов, на сегодняшний день она присутствует в мощных планшетах, телевизорах и вероятно будет интегрирована в системы управления сложными технологическими объектами. Android разрабатывается быстрыми темпами, в среднем две версии в год. Каждая новая версия получает улучшенный интерфейс, высокую производительность и множество новых пользовательских функций [1].

Одним из важных аспектов Android платформы является ее безопасность. На протяжении многих лет Android совершенствовал собственные встроенные средства защиты.

С изменением программно-аппаратной составляющей средств мобильной связи меняются способы захвата и передачи ценной информации злоумышленниками (меняется код вредоносных программ, а цели и задачи, реализующие, их остаются прежними). Метод сигнатурного анализа, реализованный в большинстве антивирусных программах, показал свою высокую эффективность при обнаружении и обезвреживании известных вирусных сигнатур, однако при появлении новых вредоносных программ, отличающихся своим поведенческим характером, их возможности становятся ограниченными. С целью увеличения качества обнаружения нового вредоносного программного обеспечения, отсутствующего в базе сигнатур, возникает необходимость и актуальность в применении новых интеллектуальных методов, опирающихся на анализ поведения вредоносного программного обеспечения. В данной работе были применены классические методы классификации программного обеспечения на основе выявленных свойств, присущих вредоносному и безопасному программному обеспечению, а также выполнен сравнительный анализ работы классических методов с машиной опорных векторов и нейросетевыми методами. Для мобильного устройства Android эти результаты являются актуальными, так как применительно к ним еще мало изучены. Анализ отечественных и зарубежных публикаций [9, 10, 11] по данной тематике показывает, что такие работы активно ведутся, однако в них отсутствует практические рекомендации, а также качественные и количественные характеристики разработанных программных проектов для систем комплексной защиты средств мобильной связи типа Android.

Постановка задачи для классических методов

Пусть X — множество объектов – программ, Y — множество примеров (virus, ok) кластеров. Классификация осуществляется на основе двух классов virus и ok. В качестве метрики выбрано Евклидово расстояние между объектами $p(x,x') = (\sum_{i=1}^{n}(x-x')^2)^{1/2}$. Задана конечная экспериментальная выборка объектов $X^m = \{x_1,...,x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами так, чтобы каждый кластер состоял из объектов, близких по метрике p, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i [4, 5, 6].

Решение задачи

Требуется определить функцию $a: X \to Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации. В качестве основных критериев были выбраны такие, которые наиболее часто используются при разборе и анализе свойств, присущих поведению вредоносного и безвредного программного обеспечения:

- Наличие txt файла содержащего списки номеров (например, smsc.txt (7921, 7923, 7924 и т.д.));
- Наличие обфускации кода;
- Класс отвечающий за шифрование File Encryptor (contex);
- Наличие класса Telephony;
- Наличие класса SmsReceiver;
- Наличие класса SmsBlockerThead;
- Наличие субкласса SmsBlockerThead;
- Наличие файла в базе сигнатур;
- Разрешения на использование сервисов системы Android.

Для решения поставленной задачи и выявления основных свойств, по которым осуществляться классификация, в данной задаче: необходимо выполнить формализацию, провести анализ и классификацию возможных состояний средства мобильной связи типа Android в условиях комплексной защиты информации.

Формализация задачи

Установка любой программы на мобильное устройство связи типа Android сопряжено с тремя основными этапами:

1. Поиск программы в интернете и при успешном обнаружении его передача через соответствующие носители на устройство.
2. Запуск файла программы с расширением apk.
3. Установка программы.

На первом этапе по запросу пользователя осуществляется поиск необходимого приложения (от игровой программы до профессионального приложения). На втором этапе, в соответствии с запрашиваемым приоритетом и необходимых сервисов, осуществляется его установка.

На третьем этапе, после подтверждения соответствующего приоритета пользователем (параллельно с его действиями в основном режиме) вредоносное программное обеспечение беспрепятственно проникает на мобильное устройство “железо” и внедряется в среду операционной системы с соответствующими правами доступа и разрешениями.

Для ликвидации указанного недостатка в настоящее время используются современные интеллектуальные методы, включающие два рубежа защиты [2]:

1. Анализ разрешений.
2. Анализ вредоносного кода.

В данной работе рассмотрим более подробно второй этап. Проведя предварительно анализ и присвоив программе тип проверяемого приложения.

Формализация вредоносного программного обеспечения осуществляется в три этапа:

1. Анализ манифеста и ресурсов приложения Android.
2. Декомпиляция.
3. Анализ кода.

Например, программа SimpleLocker имеет одну из особенностей [3].

Присутствуют ресиверы (метки):

.ServiceStarter

android.intent.action.BOOT_COMPLETED

.SDCardServiceStar

android.intent.action.ACTION_EXTERNAL_APPLICATIONS_AVAILABLE

Используются следующие основные службы:

.MainService

org.torproject.android.service.TorService

org.torproject.android.service.ITorService

org.torproject.android.service.TOR_SERVICE

Анализ конкретной программы позволил установить, что вирус работает с сетевым приложением TOR, а также с внешней памятью. Дальнейшая манипуляция с данной программой позволила выявить наиболее характерные особенности: объективная и важная информация находится в папке res/raw; кроме этого были обнаружены два замаскированных архивных файла с расширением mp3, которые являлись основой сетевой программы. Формализация и анализ работы мобильного устройства типа Android позволили на аппаратном и программном уровнях выделить основные свойства, присущие “поведенческому характеру” вредоносного программного обеспечения.

Классификация вредоносного программного обеспечения на основе выявленных признаков

Фрагмент экспериментальной выборки включает в себя перечень всех возможных разрешений на использование того или иного сервиса, который могут запрашиваться при установке программ в мольном устройстве типа Android, представленные в бинарном виде. Общее количество разрешений 162. Анализ каждого выявляемого свойства ассоциируется с профилем поведения мобильной системы, а, следовательно, с функционированием вредоносного и безвредного программного обеспечения. При этом заранее известно, что если при многообразии разрешений вирус присутствует, он соответствует значению 1 на рисунке 1 и 0 в противном случае.

	1	2	3	4	5	6	7	8	9	10	…	160	161	162	163
1	0	0	0	0	0	0	0	0	0	0	…	1	1	1	ok
2	0	0	0	0	0	0	0	0	0	0	…	0	0	0	ok
3	1	0	0	0	0	0	0	0	0	0	…	0	0	0	virus
4	1	1	1	1	1	1	1	1	1	1	…	1	1	1	virus
5	1	1	0	0	0	0	0	0	1	1	…	1	1	1	virus
6	1	1	0	1	0	0	0	0	1	0	…	0	0	0	virus
7	1	1	1	1	1	1	1	1	1	0	…	0	0	0	virus
8	1	1	0	0	0	0	0	0	1	0	…	0	0	0	virus
9	1	1	0	0	0	0	0	0	1	0	…	0	0	0	virus
10	0	0	1	1	1	1	1	1	0	0	…	1	1	1	ok
11	0	0	1	1	1	1	1	1	0	0	…	1	1	1	ok
12	0	0	1	1	1	1	1	1	0	0	…	0	0	0	ok
13	1	1	0	0	0	0	0	0	1	0	…	0	0	0	ok(?)
14	1	1	0	0	0	0	0	0	1	0	…	0	0	0	virus(?)
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
95	1	1	0	0	0	0	1	1	0	0	…	0	0	0	virus(?)
96	1	0	0	0	0	1	1	0	1	0	…	0	0	0	virus(?)
97	0	0	0	0	0	1	1	1	0	0	…	0	0	0	virus(?)
98	0	0	1	0	0	0	0	0	0	0	…	1	0	0	ok(?)
99	0	0	0	0	0	0	0	0	0	0	…	1	1	0	ok(?)
100	0	0	1	0	0	0	0	0	0	0	…	0	0	0	ok(?)

Рисунок 1. Фрагмент экспериментальных данных

В качестве критерия точности и качества работы классификатора методов (иерархической классификации, метода К-средних, машины опорных векторов и нейронных сетей) будем использовать следующую формулу:

$$ОК = \frac{ЧО \times 100\%}{ЧН}, \qquad (1)$$

где ОК – общий процент, как вредоносного, так и безвредного программного обеспечения ошибки классификации; ЧН – суммарное число наблюдений; ЧО – число ошибок классификации.

Проведя анализ каждого выявленного свойства, были составлены профили поведения как вредоносного, так и безвредного программного обеспечения. Для представления были использованы бинарные значения 1 – присутствует и 0 – отсутствует.

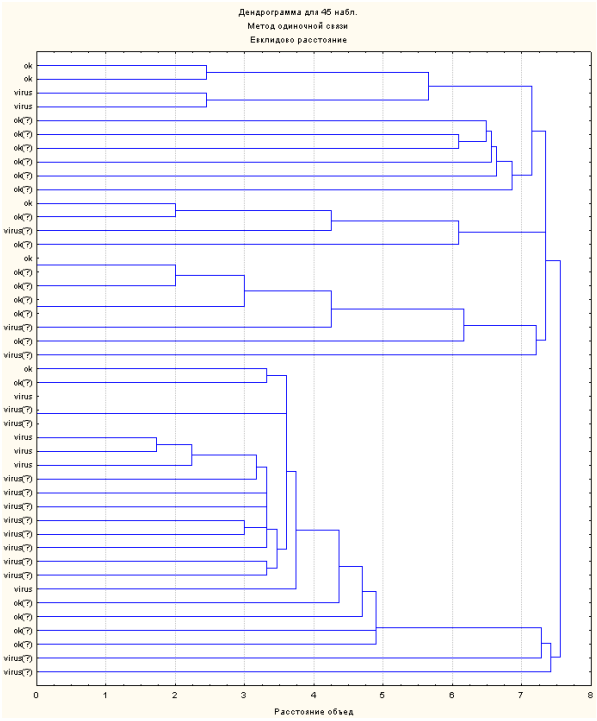

Рисунок 2. Результат работы метода иерархической классификации

Результаты работы одного из “классических” методов - иерархической классификации - приведены на рисунке 2. Анализ дендрограммы позволяет констатировать следующее: экспериментальные данные по итогам решения задачи были поделены на два множества. Каждое множество содержит как безвредные, так и вредоносные программы. Количество ошибок первого рода, когда множество безвредных программ содержит вирусы равно 22 %, ошибки второго рода, когда множество вредоносных программ функционирует исправно (вирусы не обнаружены) равно 26 %.

Результаты работы приведены в таблице 1 для метода К – средних.

Таблица 1

Результаты классификации методом К – средних

кластер 1	кластер 2
ok	ok
virus	virus
virus	virus
virus	ok
virus	ok
virus	ok
virus (?)	ok (?)
virus (?)	virus (?)
virus (?)	virus (?)
virus (?)	ok (?)
virus (?)	ok (?)
virus (?)	ok (?)
virus (?)	ok (?)
virus (?)	virus (?)
virus (?)	ok (?)
ok (?)	ok (?)
virus (?)	ok (?)
virus (?)	ok (?)
virus (?)	ok (?)
ok (?)	ok (?)
ok (?)	ok (?)
ok (?)	ok (?)
ok (?)	

В процессе проведенного эксперимента было установлено, что кластеры содержат вредоносные и безвредное программное обеспечение. Качество распознавания классических методов (иерархической классификации и К – средних) одинаковое: ошибки I рода – 26 %, ошибки II рода – 22 %, что говорит о невысокой точности работы методов. Дальнейший анализ этих методов показывает, что с увеличением количества помех (внешних и внутренних факторов в виде аддитивной или мультипликативной составляющей экспериментальной выборки) точность классификатора падает.

Применение машины опорных векторов показало его высокую эффективность в качестве классификатора:

Количество ошибок I рода – 0 %, ошибок II рода – 12 %. В результате можно утверждать, что машина опорных векторов показала лучший результат по сравнению с классическими методами.

Нейронные сети, в отличие от статистических методов многомерного классификационного анализа, базируются на параллельной обработке информации и обладают

способностью к самообучению, то есть получению обоснованного результата на основании данных, которые не встречались в процессе обучения. Эти свойства позволяют нейронным сетям решать сложные задачи, которые на сегодняшний день считаются трудноразрешимыми [8, 13, 15].

Для эффективного использования нейронных сетей необходимо наличие достаточного объема обучающей выборки, используя которую нейронную сеть можно обучить.

Выбор архитектура нейронной сети выполняется в соответствии с типом решаемой задачей. Для классификации подходят: многослойный персептрон, радиально базисная функция, вероятностные нейронные сети и сети Кохонена [10, 14].

Аргумент функции активации каждого скрытого узла сети радиальной базисной функции представляет собой Евклидову норму между входным вектором и центром радиальной функции. Аргумент функции активации каждого скрытого узла сети многослойного персептрона является скалярным произведением входного вектора и вектора синаптических весов данного нейрона.

На рисунках 3-4 указана архитектура построенных сетей с указанием активации каждого нейрона для того наблюдения, которое было задано. Интенсивность окраса нейронов соответствует их активациям, показывая визуальную индикацию активности каждой сети.

Нейронные сети, которые мы применили для классификации:

• Многослойный персептрон – указана активации каждого нейрона для наблюдения №1.

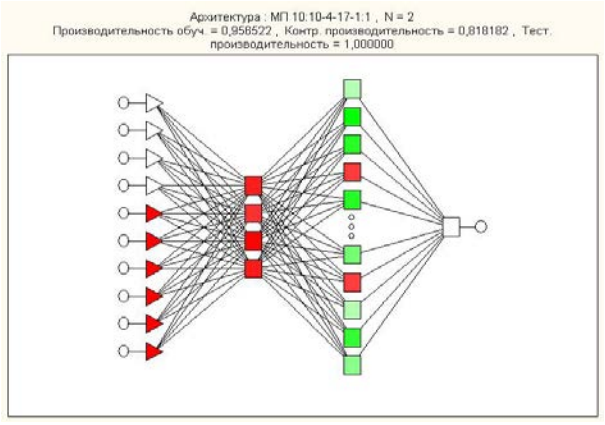

Рисунок 3. Архитектура многослойного персептрона

• Радиально базисная функция – указана активации каждого нейрона для наблюдения №1.

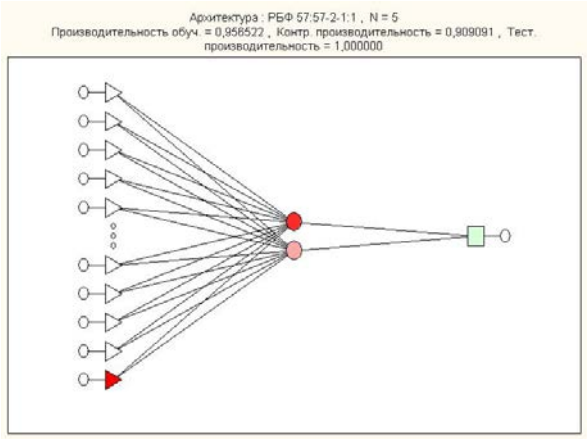

Рисунок 4. Архитектура радиально базисной функции

На рисунках также представлены: производительность обучения в целом, производительность обучения по контрольной выборке, производительность обучения по тестовой выборке, а также представлена информация о структуре нейронной сети количество входов и скрытых слоев.

Результатом работы различных архитектур нейронных сетей приведены в таблице 2.

Таблица 2

Сравнительный анализ результатов классификации нейронных сетей

№	Эталонная модель	МП (10-4-17)	РБФ (57-2-1)
1	ok	ok	ok
2	ok	ok	ok
3	virus	virus	virus
4	virus	virus	virus
5	virus	virus	virus
6	virus	virus	virus
7	virus	virus	virus
8	virus	virus	virus
9	virus	virus	virus
10	ok	ok	ok
11	ok	ok	ok
12	ok	ok	ok
13	ok	virus	ok
14	virus	virus	virus
15	virus	virus	virus
16	virus	ok	virus
17	ok	ok	ok
18	ok	ok	ok
19	ok	ok	ok
20	ok	ok	ok
21	virus	virus	virus
22	virus	virus	virus
23	virus	virus	virus
24	virus	virus	virus
25	ok	ok	ok

Продолжение табл. 2

26	ok	virus	ok
27	virus	virus	virus
28	virus	virus	virus
29	ok	ok	ok
30	virus	virus	virus
31	ok	ok	ok
32	ok	ok	ok
33	ok	ok	ok
34	ok	ok	ok
35	virus	virus	virus
36	virus	virus	virus
37	ok	ok	ok
38	ok	ok	ok
39	virus	virus	virus
40	virus	virus	virus
41	virus	virus	virus
42	ok	ok	ok
43	ok	ok	ok
44	ok	ok	virus
45	ok	ok	virus

Анализ решаемой задачи классификации на основе нейросетевых методов показывает, что лучший результат показали РБФ – сети (рисунок 5). На рисунке 5 отображены результаты проделанных экспериментов, ошибки первого и второго рода.

	Иерархической классификации		К – средних		Машина опорных векторов		МП (10-4-17-1)		РБФ (57-2-1)	
	ok	virus	ok	virus	ok	virus	ok	virus	ok	virus
Всего	23	22	23	22	23	22	23	22	23	22
Правильно	17	15	17	15	19	22	21	21	21	22
Ошибочно	5	6	6	5	4	0	2	1	2	0
% правильных	78	74	74	78	88	100	91,3	95,45	91,3	100
% ошибочных	22	26	26	22	12	0	8,69	4,5	8,69	0

Рисунок 5. Результаты работы классических и нейросетевых методов классификации

Выводы

1. В процессе сравнительного анализа классических и нейросетевых методов классификации для задачи идентификации “поведенческого характера” вируса в мобильных средствах связи типа Android установлено, что наилучшим качественными и количественными характеристиками обладают нейросетевые методы на основе архитектур РБФ и SVM – машины (машины опорных векторов).

2. Нейронная сеть находит лишь один из возможных способов разделения классов, который, не всегда является оптимальным, а машина опорных векторов реализует разделяющую поверхность, наиболее удаленную от всех разделяемых точек. Таким образом, можно предположить, что качество распознавания новых примеров у машины опорных векторов выше, чем у нейронной сети. Критерий останова для обучения нейронной сети – нулевая ошибка на обучающем множестве, а критерий останова для метода опорных векторов – близость построенной разделяющей гиперплоскости к оптимальной.

3. Основное отличие машины опорных векторов от нейронных сетей заключается в том, что для нейронных сетей количество настраиваемых коэффициентов должно априорно задаваться пользователем на основании некоторых эвристических соображений. В методе опорных векторов количество настраиваемых параметров автоматически определяется во время настройки и обычно меньше, чем число векторов в обучающей последовательности. Ненулевыми остаются коэффициенты у опорных векторов, с помощью которых строится разделяющая гиперплоскость.

4. Реализация машины опорных векторов на нейронной сети дает дополнительные возможности при решении задачи классификации, так как повышает устойчивость метода к шумам исходных данных за счет робастности нейронных сетей.

Недостатком метода опорных векторов является неустойчивость по отношению к шуму в исходных данных. Шумовые выбросы обучающей выборки будут существенным образом учтены при построении разделяющей гиперплоскости [7].

ЛИТЕРАТУРА

1 Six J. Application Security for the Android Platform. Processes, Permissions, and Other Safeguards. CA, O’Reilly Media, 2011. 2 p.

2 Жеранков С.В., Гаврилов Г.Н. Выявление вредоносных программ с использованием современного интеллектуального метода на этапе установки // XIII Международная научно-практическая конференция: Научные перспективы XXI века. Достижения и перспективы нового столетия, Новосибирск, 10–11 июль 2015 г. Новосибирск: Изд-во Международный научный Институт “Educatio”, 2015. С. 134–138.

3 Бояркин А., Набиев Н. Анализ Simplelocker-a - вируса-вымогателя для Android [Электронный ресурс] М.: ТМ, 2014. Режим доступа: http://habrahabr.ru/company/pentestit/blog/237207/ (23.08.2015).

4 Воронцов К. Методы кластеризации [Электронный ресурс]. Режим доступа: http://www.MachineLearning.ru/wiki?title=User:Vokov (26.08.2015).

5 Кластерный анализ (кластеризация) [Электронный ресурс]. Режим доступа: http://statistica.ru/glossary/general/klasternyy-analiz-klasterizatsiya/ (01.09.2015).

6 Котельников Е., Козвонина А. Параллельная реализация машины опорных векторов с использованием методов кластеризации [Электронный ресурс]. Режим доступа: http://ict.informika.ru/vconf/files/11508.pdf (03.09.2015).

7 Любимов Н., Михеев Е., Лукин А. Сравнение алгоритмов кластеризации в задаче диктора [Электронный ресурс]. Режим доступа: http://www.researchgate.net/publication/267690636 (03.09.2015).

8 Черезов Д., Тюкачев Н. Обзор основных методов классификации и кластеризации данных [Электронный ресурс] // Вестник ВГУ. 2009. Режим доступа: http://www.vestnik.vsu.ru/pdf/analiz/2009/02/2009-02-05.pdf (05.09.2015).

9 Sanz B., Santos I., Nieves J., Laorden C. et al. MADS: Malicious android applications detection through string analysis [Electronic recourse] // Network and System Security, Springer Berlin Heidelberg. 2011. V. 5. Available at: http://www.researchgate.net/publication/256194745_MADS_Malicious_Android_Applications Detection_through_String_Analysis (Accessed 08 March 2015).

10 Fan Yuhui, Xu Ning The Analysis of Android Malware Behaviors [Electronic recourse] // International Journal of Security and Its Applications. 2015. V. 9. № 3. Available at: http://www.sersc.org/journals/IJSIA/vol9_no3_2015/25.pdf (Accessed 08 March 2015).

11 Arp D., Spreitzenbarth M., Hubner M., Gascon H. et al. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket [Electronic recourse] // NDSS Symposium 2014, Switzerland. 2014. V. 4. № 1. Available at: https://user.informatik.unigoettingen.de/~krieck/docs/2014-ndss.pdf (Accessed 08 March 2015).

12 Донцова Л., Донцов Е. Сравнение метода опорных векторов и нейронной сети при прогнозировании банкротства предприятий. [Электронный ресурс]. Режим доступа: http://urf.podelise.ru/docs/1100/index-78995.html (08.09.2015).

13 Нейронные сети [Электронный ресурс]. Режим доступа: http://www.statlab.kubsu.ru/sites/project_bank/nural.pdf (14.11.2015).

14 Боровиков В.П. Нейронные сети. Statistica Neural Networks. Методология и технологии современного анализа данных. М.: Физматлит, 2009. 392 с.

15 Нейронные сети [Электронный ресурс]. Режим доступа: http://www.statsoft.ru/home/textbook/modules/stneunet.html (15.11.2015).

REFERENCES

1 Six J. Application Security for the Android Platform. Processes, Permissions, and Other Safeguards. CA, O'Reilly Media, 2011. 2 p.

2 Zherankov S.V. Gavrilov G.N. Identify malware using advanced predictive method during installation. XIII Mezhdunarodnaya nauchno-prakticheskaya konferentsiya. Nauchnye perspektivy XXI veka [XIII International Scientific-Practical Conference: Scientific Perspectives XXI century. Achievements and prospects of the new century. Publishing House of International Scientific Institute "Educatio"]. 2015. pp. 134-138. (In Russ.).

3 Boyarkin A., Nabiyev N. Analiz Simplelocker-a – virusa-vymogatelya Android [Analysis Simplelocker-a - virus-extortionist for Android. M.: TM, 2014]. Available at: http://habrahabr.ru/company/pentestit/blog/237207/ (Accessed 23 October 2015). (In Russ.).

4 Vorontsov K. Metody klasterizatsii [Clustering methods]. Available at: http://www.MachineLearning.ru/wiki?title=User:Vokov (Accessed 26 October 2015). (In Russ.).

5 Klasternyi analiz [Cluster analysis (clustering)]. Available at: http://statistica.ru/glossary general/klasternyy-analiz-klasterizatsiya/ (Accessed 23 October 2015). (In Russ.).

6 Kotelnikov E., Kozvonina A. Parallel'naya realizatsiya mashiny opornykh vektorov s ispol'zovaniem metodov klasterizatsii [Parallel implementation of support vector machines using clustering methods]. Available at: http://ict.informika.ru/vconf/files/11508.pdf (Accessed 3 October 2015). (In Russ.).

7 Lyubimov N. Mikheyev E. Lukin A. Sravnenie algoritmov klasterizatsii v zadache diktora [Comparison of clustering algorithms in the problem of the speaker. Available at: http://www.researchgate.net/publication/267690636] (Accessed 3 October 2015). (In Russ.).

8 Cherezov D., Tyukachev N. Obzor osnovnykh metodov klassifikatsii i klasterizatsii dannykh [Overview main methods of data classification and clustering. Voronezh Bulletin MAD. 2009. 2014]. Available at: http://www.vestnik.vsu.ru/pdf/analiz/2009/02/2009-02-05.pdf (Accessed 5 October 2015). (In Russ.).

9 Sanz B., Santos I., Nieves J., Laorden C. et al. MADS: Malicious android applications detection through string analysis. Network and System Security, Springer Berlin Heidelberg, 2011, vol. 5, no. Available at: http://www.researchgate.net/publication/256194745_MADS_Malicious_Android_Applicat

ions_Detection_through_String_Analysis (Accessed 08 March 2015).

10 Fan Yuhui, Xu Ning The Analysis of Android Malware Behaviors. International Journal of Security and Its Applications, Australia, 2015, vol. 9, no. 3. Available at: http://www.sersc.org/journals/IJSIA/vol9_no3_2015/25.pdf (Accessed 08 March 2015).

11 Arp D., Spreitzenbarth M., Hubner M., Gascon H. et al. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. NDSS Symposium 2014, Switzerland, 2014, vol. 4, no. 1. Available at: https://user.informatik.uni-goettingen.de/~krieck/docs/2014-ndss.pdf (Accessed 08 March 2015).

12 Dontsova L., Dontsov E. Sravnenie metoda opornykh vektorov i neironnoi seti pri prognozirovanii [Comparison of the support vector machine and the neural network in predicting bankruptcy]. Available at: http://urf.podelise.ru/docs/1100/index-78995.html (Accessed 8 October 2015). (In Russ.).

13 Neironnye seti [Neural networks]. Available at: http://www.statlab.kubsu.ru/sites/project_bank/nural.pdf (Accessed 14 November 2015). (In Russ.).

14 Borovikov V.P. Neironnye seti [Neural networks. Statistica Neural Networks. Methodology and technology of modern data analysis. Classical and neural network classification methods]. Moscow, FIZMATLIT, 2009. 392 p. (In Russ.).

15 Neironnye seti [Neural networks]. Available at: http://www.statsoft.ru/home/textbook/modules/stneunet.html (Accessed 15 November 2015). (In Russ.).