Работа частично поддержана Белорусским республиканским фондом фундаментальных исследований, грант Ф08М-097.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Беляев, Б.И. Оптическое дистанционное зондирование / Б.И. Беляев, Л.В. Катковский – Минск: БГУ, 2006. – 455 с.
2. Chao, K., Y.R. Chen, and M. S. Kim. Machine vision technology for agricultural applications // Elsevier science transactions on computers and electronics in agriculture. – 2002. – Vol. 36.– P. 173–191.
3. Федер, Е. Фракталы / Е. Федер. – М.: Мир, 1991. – 254 с.
4. Ganchenko, V. Joint segmentation of Aerial Photographs with the Various Resolution / V. Ganchenko, A. Petrovsky, B. Sobkoviak // Proc. of the 5th Int. Conf. on Neural Networks and Artificial Intelligence, ICNNAI 2008, May 27-30, 2008, Minsk, Belarus – Minsk, 2008. – P. 177–181.
5. Francis S.Hill. Computer Graphics. // Macmillan Publishing Company. – 1990.
6. Foley, van Dam, Feiner, Hughes. Computer Graphics: Principles and Practice. – Addison-Wesley Publishing Company, 1990.
7. Haralick R.M., Shanmugam K., Dinstein I. Textural Features for Image Classification // IEEE Transactions on Systems, Man and Cybernetics. – 1973. – No.6. – P. 610-621.
8. Старовойтов, В.В. Локальные геометрические методы цифровой обработки и анализа изображений / В.В. Старовойтов. – Минск: Институт технической кибернетики НАН Беларуси, 1997. – 284 с.
9. Городецкий, В.И. Многоагентные системы (обзор) / В.И. Городецкий, М.С. Грушинский, А.В. Хабалов // Новости искусственного интеллекта. – 1997. – №1. – С. 64–117.
10. Jennings, N.R. A roadmap of agent research and development / N.R. Jennings, K. Sycara, M. Wooldridge. // Autonomous Agents and Multi-Agent Systems. – Kluwer, 1998. – P. 275–306.
11. Nwana, H.S. Software Agent Technologies / H.S. Nwana, M. Wooldridge // British Telecommunications Technology Journal. – 1996. – №. 14 (4). – P. 16–27.
12. Nwana, H.S. Software Agents: An Overview. / H.S. Nwana // The Knowledge Engineering Review. – 1996. – №11. – P. 205–244.
13. Otwagin, A.A Multiagent System for Reliable and Efficient Parallel Computing / A. Otwagin / Neural Networks and Artificial Intelligence (ICNNAI-2008) // Proceedings of the Fifth International Conference (27-30 May, 2008, Minsk, Belarus). – Minsk: Propilei, 2008. – P. 46–50.
14. Doudkin A. Special areas detection using fractal and textural characteristics of high resolution images / A. Doudkin, V. Ganchenko, A. Petrovsky, B. Sobkowiak // Proceedings of the 9th International Conference on Pattern Recognition and Information Processing (PRIP'2009), May, 19-21, Minsk, Belarus – Minsk: Publ/ center of BSU, 2009. – P. 137–139.
15. Doudkin A. Potato disease detection using color leaves characteristics / A. Doudkin, V. Ganchenko, A. Petrovsky, M. Vatkin // Proceedings of the 9th International Conference on Pattern Recognition and Information Processing (PRIP'2007), May, 22–24, Minsk, Belarus – Minsk: UIIP of NASB, 2007. – Vol. 2. – P. 83–86.

Материал поступил в редакцию 15.09.09

GANCHENKO V.V. Program system of areas detection on high resolution aerospace images

Description of special areas detection program system in high spatial resolution images is represented in the paper and the result of the system work is illustrated on an example of agricultural fields images.

УДК 004.056.57:032.26

Безобразов С.В., Головко В.А.

НЕЙРОСЕТЕВАЯ ИСКУССТВЕННАЯ ИММУННАЯ СИСТЕМА ДЛЯ ОБНАРУЖЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ: ПРИНЦИПЫ ПОСТРОЕНИЯ

Введение. Защита собственных информационных ресурсов становится одной из главных задач современного государства. Число компьютерных преступлений увеличивается ежегодно на 30–40 процентов [1]. С каждым годом киберпреступления охватывают все новые и новые сферы: разработка и внедрение вредоносных программ, кража конфиденциальной информации, взлом информационных ресурсов, организация сетевых атак и т.д. Наибольшую угрозу для информации несут вредоносные программы, разрабатываемые злоумышленниками с целью уничтожения или кражи информации. Их количество из года в год увеличивается, а ущерб, наносимый компьютерными вирусами, составляет, по некоторым подсчетам, миллиарды долларов в год [2].

На сегодняшний день никто не может быть полностью уверен в защите собственной компьютерной системы от вторжения вредоносных программ. Как показала практика, традиционный подход в области обнаружения вредоносных программ, основанный на сигнатурном анализе [3, 4], не приемлем для обнаружения неизвестных компьютерных вирусов. Для поддержания должного уровня защиты пользователи вынуждены постоянно и своевременно обновлять антивирусные базы. Однако задержка в ответной реакции со стороны антивирусных компаний на появление новой вредоносной программы (ее обнаружение и создание сигнатуры) может варьироваться от нескольких часов до нескольких суток. За это время вредоносные программы способны нанести непоправимый ущерб.

В свою очередь, эвристические алгоритмы [5, 6, 7], разработанные специально для обнаружения неизвестных вредоносных программ, характеризуются высоким уровнем возникновения ошибок первого и второго родов (ложные срабатывания).

Современные исследования в области защиты информации направлены на создание таких методов и алгоритмов защиты, которые были бы способны обнаруживать и нейтрализовывать неизвестные вредоносные программы, и таким образом не только повысить уровень компьютерной безопасности, но и избавить пользователя от постоянных обновлений антивирусного ПО или его модулей.

Предпосылкой для создания эффективных антивирусных систем является развитие искусственных иммунных систем [8, 9] и нейросетевых технологий [10, 11, 12], которые имеют биологические основы. Способность таких систем к обучению, обобщению результатов и самоорганизации позволяет создавать на их базе интеллектуальные системы защиты информации.

В данной статье предложены принципы построения нейросетевой искусственной иммунной системы для обнаружения вредоносных программ, позволяющей обнаруживать новые компьютерные вирусы. Такая система состоит из популяции нейросетевых иммунных детекторов, которые применяются для обнаружения вредоносных программ и нейросетевого классификатора, предназначенного для классификации обнаруженных вредоносных программ (рис. 1).

Безобразов Сергей Валерьевич, *старший преподаватель кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*

Головко Владимир Адамович*, д.т.н., профессор, зав. кафедрой интеллектуальных информационных технологий Брестского государственного технического университета.*

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

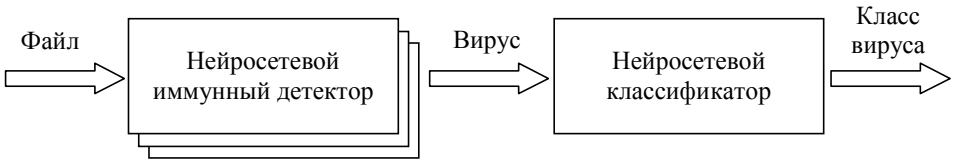

Рис. 1. Нейросетевой модуль искусственной иммунной системы для обнаружения и классификации вредоносных программ

Алгоритм построения нейросетевой искусственной иммунной системы для обнаружения вредоносных программ. Основные принципы построения искусственной иммунной системы нами были подробно представлены в [13–17]. Основываясь на разработанных алгоритмах, рассмотрим процесс построения искусственной иммунной системы на основе нейронных сетей.

На первом этапе генерируется начальная популяция иммунных детекторов, каждый из которых представляет собой искусственную нейронную сеть. Представим нейросетевой иммунный детектор в виде черного ящика, который имеет n-входов и два выхода (рис. 2).

Рис. 2. Нейросетевой иммунный детектор

Выходные значения детектора формируются после подачи всех образов на него в соответствии со следующими выражениями

$$Z_1 = \begin{cases} 1, & \textit{если чистый файл} \\ 0, & \textit{иначе.} \end{cases}$$
$$Z_2 = \begin{cases} 1, & \textit{если вирус} \\ 0, & \textit{иначе.} \end{cases} \quad (1)$$

Набор из чистых файлов и вредоносных программ образуют обучающую выборку для нейросетевых детекторов. Присутствие вируса или его сигнатуры при обучении позволяет обученным иммунным детекторам находить разницу между неинфицированными файлами и компьютерными вирусами. Очевидно, что чем больше разнообразных файлов присутствуют в обучающей выборке, тем разнообразнее будут иммунные детекторы. Желательно также иметь представителей всех типов вредоносных программ – черви, троянские программы, макровирусы и т.д. [3]. Однако это необязательное условие, потому что вредоносные программы структурно (по набору команд) отличаются от неинфицированных файлов, так как подразумевают деструктивные функции, что влияет на решение иммунного детектора при сканировании файла. Нейронная сеть обучается путем обучения с учителем [10], т.е. мы указываем искусственной нейронной сети, где данные из чистых файлов, а где из вредоносных программ. Механизм обучения детектора представлен на рисунке 3.

Как уже отмечалось, для обучения нейросетевых иммунных детекторов выбираются два класса объектов: «чистые» и вредоносные программы. Для формирования объектов чистого класса выбираются файлы из числа системных утилит операционной системы Microsoft Windows (на рисунке 3 это: dwwin.exe, regedit.exe, taskman.exe, autoras.exe). Для формирования класса вредоносных программ используются компьютерные вирусы (на рис. 3 это – lovesan.vir).

Пусть T – множество чистых файлов, а F – множество вредоносных файлов. Из них случайным образом формируется множество входных образов для обучения i-го детектора.

$$X_i = \begin{bmatrix} X_i^1 \\ X_i^2 \\ \dots \\ X_i^L \end{bmatrix} = \begin{bmatrix} X_{i1}^1 & X_{i2}^1 & \dots & X_{in}^1 \\ X_{i1}^2 & X_{i2}^2 & \dots & X_{in}^2 \\ \dots & & & \\ X_{i1}^L & X_{i2}^L & \dots & X_{in}^L \end{bmatrix}. \quad (2)$$

Соответственно, множество эталонных образов

$$I_i = \begin{bmatrix} I_i^1 \\ I_i^2 \\ \dots \\ I_i^L \end{bmatrix} = \begin{bmatrix} I_{i1}^1 & I_{i2}^1 \\ I_{i1}^2 & I_{i2}^2 \\ \dots & \\ I_{i1}^L & I_{i2}^L \end{bmatrix}, \quad (3)$$

где L – размерность обучающей выборки.

Эталонные выходные значения для i-го детектора формируются следующим образом:

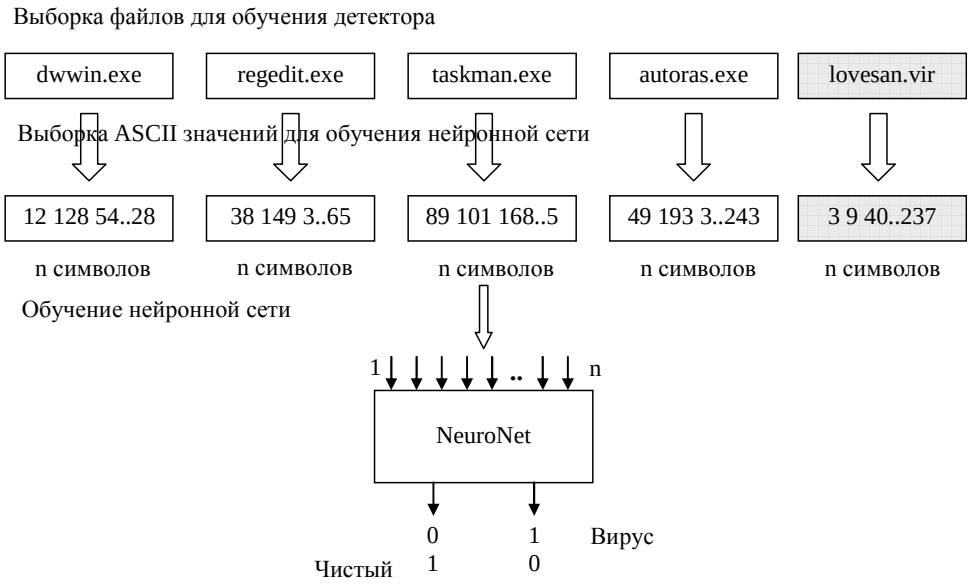

Рис. 3. Механизм обучения иммунного детектора на основе нейронной сети

$$I_{i1}^k = \begin{cases} 1, & \text{если } X_i^k \in T \\ 0, & \text{иначе.} \end{cases}$$
$$I_{i2}^k = \begin{cases} 1, & \text{если } X_i^k \in F \\ 0, & \text{иначе.} \end{cases} \quad (4)$$

Обучение каждого детектора осуществляется с целью минимизации суммарной квадратичной ошибки детектора. Суммарная квадратичная ошибка i-го детектора определяется следующим образом:

$$E_i = \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{2}(Z_{ij}^k - I_{ij}^k)^2, \quad (5)$$

где Z_{ij}^k – значение j-го выхода i-го детектора при подаче на вход его k-го образа.

Величина суммарной квадратичной ошибки характеризует приспособленность детектора к обнаружению вредоносных файлов. Чем меньше ее значение, тем больше приспособленность детектора. Поэтому величину суммарной квадратичной ошибки можно использовать для отбора лучших детекторов.

Набор обученных нейронных сетей образует популяцию иммунных детекторов, которые циркулируют в компьютерной системе и производят обнаружение вредоносных программ. Наличие разнообразных файлов для обучения и элемента случайности в формировании входных векторов дает возможность получить большое количество различных по своей структуре иммунных детекторов. В процессе сканирования неизвестного файла нейронная сеть идентифицирует неизвестный образ, в результате чего иммунный детектор принимает решение о принадлежности файла к классу вредоносных программ или к классу чистых файлов.

Общий алгоритм построения и функционирования нейросетевой иммунной системы можно представить в виде следующей последовательности:

1. Генерация начальной популяции иммунных детекторов, каждый из которых представляет собой искусственную нейронную сеть со случайными синаптическими связями:

$$D = \left\{D_i, \quad i = \overline{1,r}\right\}, \quad (6)$$

где D_i – i-й нейросетевой иммунный детектор,

r – общее количество детекторов.

2. Обучение сформированных иммунных нейросетевых детекторов. Обучающая выборка формируется случайным образом из совокупности чистых файлов (как правило, это разнообразные системные утилиты операционной системы) и из совокупности вредоносных программ, или их сигнатур. Эталонные выходные значения нейронной сети формируются соответственно формуле 4.

3. Отбор (селекция) нейросетевых иммунных детекторов на тестовой выборке. На данной итерации уничтожаются те детекторы, которые оказались неспособны к обучению, и детекторы, в работе которых, наблюдаются различные недостатки (например, ложные срабатывания). Для этого каждый детектор проверяется на тестовой выборке. В результате для каждого детектора определяется значение квадратичной ошибки E_i (формула 5).

Селекция детектора производится следующим образом:

$$D_i = \begin{cases} 0, & \text{если } E_i \neq 0 \\ D_i, & \text{иначе.} \end{cases} \quad (7)$$

где 0 обозначает операцию уничтожения детектора.

4. Каждый детектор наделяется временем жизни и случайным образом выбирает файл для сканирования из совокупности файлов, которые он не проверял.

5. Сканирование каждым детектором выбранного файла, в результате которого определяются выходные значения детекторов Z_{i1}, Z_{i2}, $i=1,r$.

6. Если i-й детектор не обнаружил вирус в сканируемом файле, т.е. $Z_{i1}=1$ и $Z_{i2}=0$, то он выбирает следующий файл для сканирования. Если время жизни i-го детектора закончилось, то он уничтожается и вместо него генерируется новый детектор.

7. Если i-й детектор обнаружил вирус в сканируемом файле, т.е. $Z_{i1}=0$ и $Z_{i2}=1$, то подается сигнал об обнаружении вредоносного файла и осуществляются операции клонирования и мутации соответствующего детектора. Операция мутации заключается в дополнительном обучении детекторов-клонов на обнаруженном вредоносном файле. В результате создается совокупность детекторов, настроенных на обнаруженную вредоносную программу

$$D_i = (D_{i1}, D_{i2}, \ldots, D_{in}). \quad (8)$$

8. Отбор клонированных детекторов, которые являются наиболее приспособленными к обнаружению вредоносной программы. Если $E_{ij} > E_i$, то детектор прошел отбор. Здесь E_{ij} – суммарная квадратичная ошибка j-го клона i-го детектора, которая вычисляется на вредоносном файле.

9. Детекторы-клоны осуществляют сканирование файлового пространства компьютерной системы до тех пор, пока не произойдет уничтожение всех проявлений вредоносной программы.

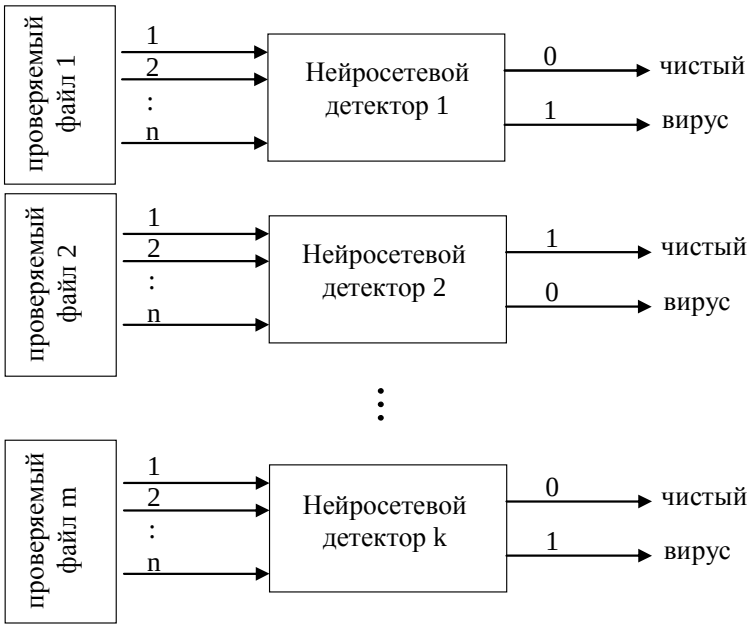

Рис. 4. Механизм функционирования иммунных детекторов на основе нейронных сетей

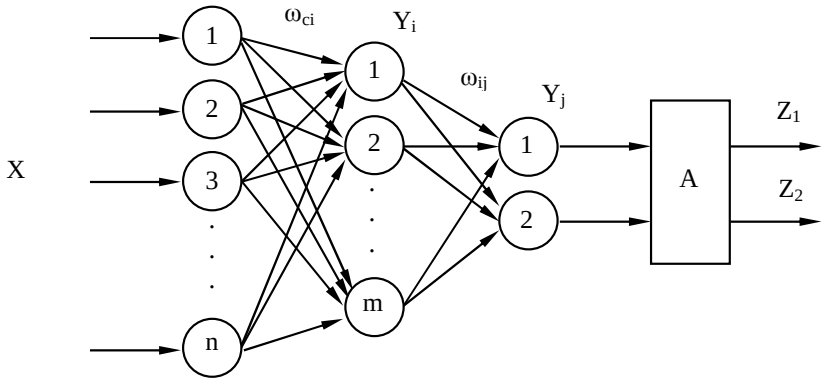

Рис. 5. Нейросетевой иммунный детектор

10. Формирование детекторов иммунной памяти. На этой итерации определяются нейросетевые иммунные детекторы, показавшие наилучшие результаты при обнаружении присутствующего в компьютерной системе вируса. Детекторы иммунной памяти находятся в системе достаточно длительное время и обеспечивают защиту от повторного заражения.

Рисунок 4 демонстрирует работу совокупности иммунных детекторов, построенных при применении нейронных сетей.

Отметим основные отличия предложенного алгоритма от существующих. В данном случае каждый нейросетевой иммунный детектор является полностью самостоятельным объектом, т.е. сам выбирает себе область сканирования. Для этого он получает список файлов, хранящихся на жестком диске, и случайным образом выбирает файл из списка для его проверки. После проверки одного файла детектор переходит к следующему файлу, также выбранному случайным образом из существующего списка. Сканирование файлов нейросетевым иммунным детектором продолжается до тех пор, пока детектор не обнаруживает вредоносную программу, либо до истечения времени, отведенного для функционирования данного детектора.

Популяция нейросетевых иммунных детекторов обеспечивает достаточную область покрытия сканирования файлов на жестком диске для своевременного обнаружения проникшего потенциального компьютерного вируса.

Таким образом, соблюдается принцип децентрализации системы безопасности, построенной на основе комбинации методов нейронных сетей и искусственных иммунных систем, что значительно повышает отказоустойчивость и защищенность системы в целом.

2. Структура и обучение нейросетевого иммунного детектора. В предыдущем разделе была предложена организация и функционирование нейросетевой искусственной иммунной системы для обнаружения вредоносных программ. В данном разделе рассматривается структура и обучение иммунного детектора, в основе которого лежит нейронная сеть. Основной задачей нейросетевого иммунного детектора является разделение пространства входных образов на два класса: не вредоносный (чистый) класс и вредоносный (вирусный) класс. Как было отмечено ранее, главными недостатками генетического иммунного детектора являются значительные временные и вычислительные затраты, а также недостаточная способность к обнаружению разнообразных компьютерных вирусов. Поэтому предлагается использовать нейросетевой подход к построению иммунного детектора, который обладает способностью к обучению и обобщению результатов обучения при подаче на вход детектора неизвестных образов.

Рассмотрим выбор класса нейронной сети, лежащей в основе нейросетевого иммунного детектора. В процессе циркуляции НИД происходит их непрерывная эволюция, путем уничтожения старых и формирования новых детекторов. После генерации новых детекторов происходит процесс их обучения, трудоемкость которого пропорциональна размерности обучающей выборки. Поэтому, для увеличения быстродействия нейросетевой искусственной иммунной системы необходимо выбрать такой класс нейронной сети, который характеризуется минимальным размером обучающей выборки. Рассмотрим многослойный персептрон [10, 11], который состоит из n нейронов распределительного слоя, m нейронов скрытого слоя и 2 нейронов выходного слоя. Общее количество настраиваемых параметров (весовых коэффициентов и пороговых значений) в такой сети определяется следующим образом:

$$V = m \cdot (n + 3) + 2 . \qquad (9)$$

Для хорошей классификации размер обучающей выборки должен определяться в соответствии со следующим выражением [11]:

$$L \approx \frac{V}{\varepsilon}, \qquad (10)$$

где ε – допустимая точность классификации.

Пусть n = 128, m = 10 и ε = 0,1. Тогда L ≈13120.

Аналогичный результат можно получить для мультирекуррентных нейронных сетей [10, 11].

Рассмотрим аналогичную сеть встречного распространения [10, 11] с идентичным количеством нейронных элементов в слоях. В скрытом слое будем использовать нейронные элементы Кохонена. В этом случае нет жестких требований к размерности обучающей выборки. Достаточно, чтобы размер обучающей выборки был следующим:

$$L \geq 2 \cdot m. \qquad (11)$$

Поэтому выберем в качестве основы нейросетевого иммунного детектора нейронную сеть встречного распространения.

На рисунке 5 изображена архитектура нейросетевого иммунного детектора, который состоит из трех слоев нейронных элементов и арбитра. На вход такого детектора в режиме функционирования подаются фрагменты проверяемого файла, которые формируются в соответствии с методом скользящего окна. Первый слой нейронных элементов является распределительным. Он распределяет входные сигналы на нейронные элементы второго (скрытого) слоя. Количество нейронных элементов распределительного слоя равняется размерности скользящего окна.

Второй слой состоит из нейронов Кохонена, которые используют конкурентный принцип обучения и функционирования в соответствии с правилом «победитель берет все» [10, 11].

Третий слой состоит из двух линейных нейронных элементов, которые используют линейную функцию активации. Арбитр осуществляет процедуру окончательного решения о принадлежности сканируемого файла к вирусному или чистому классу.

Рассмотрим выбор количества нейронов в слое Кохонена. Нейронный слой Кохонена осуществляет кластеризацию входного пространства образов, в результате чего образуются кластеры различных образов, каждому из которых соответствует свой нейронный элемент. Количество нейронов слоя Кохонена равняется m. Причем

$$m = p + r , \qquad (12)$$

где p – количество первых нейронов слоя Кохонена, которые соответствуют классу чистых программ,

r – количество последних нейронов слоя Кохонена, активность которых характеризует класс вредоносных программ.

При обучении нейросетевых иммунных детекторов используется обучающая выборка, состоящая из 80% образов чистого класса и из 20% образов вредоносного класса. Таким образом, соотношение файлов в обучающей выборке равняется четыре к одному. Данное соотношение было получено экспериментальным путем и показало наилучшие результаты (таблица 1).

Таблица 1. Пространство выходных значений арбитра

Имя файла	5/1	4 / 1	3 / 1	2 / 1	1 / 1
Backdoor.Agent	***чистый***	вирус	вирус	***чистый***	***чистый***
Backdoor. Agobot	вирус	вирус	вирус	вирус	вирус
E-Worm.Bozori	вирус	вирус	вирус	вирус	вирус
E-Worm.Zafi	вирус	вирус	вирус	вирус	вирус
E-Worm.Mydoom	вирус	вирус	вирус	вирус	вирус
E-Worm.NetSky	вирус	вирус	вирус	вирус	вирус
Exploit.DebPloit	вирус	вирус	***чистый***	***чистый***	вирус
Net-Worm.Lovesan	вирус	вирус	вирус	вирус	вирус
Net-Worm.Mytob	вирус	вирус	вирус	вирус	вирус
Trojan.Bagle	вирус	вирус	вирус	вирус	вирус
Trojan.Daemonize	вирус	вирус	вирус	вирус	вирус
Trojan.LdPinch	***чистый***	вирус	вирус	вирус	вирус
Virus.Gpcode	вирус	вирус	вирус	вирус	вирус
Virus.Hidrag	вирус	вирус	вирус	вирус	вирус
cacls.exe	чистый	чистый	чистый	***вирус***	***вирус***
ctfmon.exe	чистый	чистый	чистый	чистый	чистый
dbexplor.exe	***вирус***	чистый	чистый	чистый	чистый
dcomcnfg.exe	чистый	чистый	чистый	***вирус***	***вирус***
diskcopy.com	чистый	чистый	чистый	чистый	***вирус***
dllhost.exe	***вирус***	чистый	чистый	чистый	чистый
etm70.exe	чистый	чистый	чистый	чистый	чистый
notepad.exe	чистый	чистый	чистый	чистый	чистый
soundman.exe	чистый	чистый	чистый	чистый	чистый
taskman.exe	чистый	чистый	***вирус***	чистый	***вирус***
uninlib.exe	чистый	чистый	чистый	***вирус***	***вирус***

В проведенных экспериментах генерировалось 5 совокупностей нейросетевых иммунных детекторов, состоящих из 100 детекторов, которые затем проходили стадию обучения и отбора. Для первой популяции нейросетевых иммунных детекторов использовалась обучающая выборка, состоящая из соотношения образов чистого класса к образам вредоносного класса 5/1, для второй популяции – 4/1, для третьей популяции – 3/1, для четвертой – 2/1, для пятой популяции – 1/1. После обучения и отбора детекторы сканировали тестовые файлы. Результаты работы детекторов приведены в таблице 1.

Как видно из таблицы 1, наилучший результат показали детекторы, для обучения которых использовалась выборка, состоящая из 80% образов чистого класса и 20% образов вредоносного класса.

Пусть A – количество фрагментов выбираемых из каждого файла для обучения и имеется 4 чистых файла и один вредоносный. Тогда обучающая выборка состоит из $4A$ фрагментов, относящихся к чистому классу, и A фрагментов, характеризующих вредоносный класс. Поэтому соотношение между количеством нейронов в слое Кохонена, которые характеризуют различные классы, должно быть кратным соотношению четыре к одному

$$\frac{p}{r} = \frac{i}{i} \cdot \frac{4}{1}, \tag{13}$$

где i = 1,2…

Так, например, при i=1

$$p = 4, r = 1, \tag{14}$$

а при $i = 2$

$$p = 8, r = 2. \tag{15}$$

Отсюда следует, что алгоритм формирования обучающей выборки состоит из следующих шагов:

1. Формируется совокупность чистых и вирусных файлов.
2. Из сформированной выборки случайным образом выбираются четыре чистых и один вирусный файл.
3. Из каждого файла случайным образом выбираются A фрагментов длиной n, в результате чего образуется обучающая выборка размерностью $L = 5A$.

Для обучения нейронов слоя Кохонена используется контролируемое конкурентное обучение [10, 11]. При таком обучении весовые коэффициенты нейрона-победителя модифицируются только тогда, когда происходит корректная классификация входного образа, т.е. входной образ соответствует заданному множеству нейронов в слое Кохонена. Так как в слое Кохонена используется p нейронов для чистых входных образов и r нейронов для вредоносных входных образов, то корректная классификация происходит, если при подаче на вход сети чистого фрагмента победителем является один из первых p нейронов слоя Кохонена. Аналогичным образом корректная классификация происходит, если при подаче на вход сети вирусного фрагмента, победителем является один из r последних нейронов слоя Кохонена. В остальных случаях происходит некорректная классификация.

Пусть P и J характеризуют соответственно чистый и вредоносный файл. Тогда правило корректной классификации можно представить в виде следующей импликации:

$$\begin{aligned} P \wedge k = 1,2 \ldots p \rightarrow T, \\ J \wedge k = p+1, r \rightarrow T, \end{aligned} \tag{16}$$

где T обозначает корректную классификацию.

При корректной классификации весовые коэффициенты нейрона-победителя усиливаются

$$\omega_{ck}(t+1) = \omega_{ck}(t) + \gamma(X_c - \omega_{ck}(t)), \tag{17}$$

а при некорректной классификации ослабляются:

$$\omega_{ck}(t+1) = \omega_{ck}(t) - \gamma(X_c - \omega_{ck}(t)), \tag{18}$$

где γ – шаг обучения.

Алгоритм обучения слоя Кохонена состоит из следующих шагов:

1. Случайная инициализация весовых коэффициентов нейронов слоя Кохонена.
2. Подается входной образ из обучающей выборки на нейронную сеть и производятся следующие вычисления:

a) вычисляется Евклидово расстояние между входным образом и весовыми векторами нейронных элементов слоя Кохонена

$$\begin{aligned} D_i &= \left| X - \omega_i \right| = \\ &= \sqrt{(X_1 - \omega_{1i})^2 + (X_2 - \omega_{2i})^2 + (X_n - \omega_{ni})^2}, \end{aligned} \tag{19}$$

где $i = \overline{1,m}$.

b) определяется нейронный элемент-победитель с номером k

$$D_k = \min_j D_j \tag{20}$$

Таблица 2. Результаты сравнительного анализа обнаружения

Имя файла	Антивирус Касперского (актуал. базы)	Антивирус Касперского (устар. базы)	NOD32 (эвристическ. анализатор)	ИИС (на основе 4-х детекторов)
Backdoor.Win32.Agent.lw	Backdoor	OK	OK	OK
Backdoor.Win32.Agobot	Backdoor	Backdoor	Win32/Agobot	Вирус
Email-Worm.BAT.Maddas	Email-Worm	Email-Worm	OK	Вирус
Email-Worm.JS.Gigger	Email-Worm	Email-Worm	OK	Вирус
Email-Worm.VBS.Loding	Email-Worm	Email-Worm	OK	Вирус
Email-Worm.Win32.Zafi.d	Email-Worm	OK	NewHeur_PE	Вирус
Net-Worm.Win32.Bozori.a	Net-Worm	OK	Win32/Bozori	Вирус
Net-Worm.Win32.Mytob.a	Net-Worm	OK	Win32/Mytob	Вирус
Trojan-Downl.JS.Psyme.y	Trojan	OK	OK	Вирус
Trojan-Downl.Win32.Bagle	Trojan	OK	Win32/Bagle	Вирус
Trojan-Proxy.Daemonize	Trojan	Trojan	OK	OK
Trojan-Proxy.Mitglieder	Trojan	Trojan	Win32/Trojan	Вирус
Trojan-Proxy.Win32.Agent	Trojan	Trojan	OK	Вирус
Trojan-PSW.LdPinch	Trojan	Trojan	Win32/PSW	Вирус
Virus.Win32.Gpcode.ac	Virus.Win32	OK	OK	Вирус
Exploit.Win32.DebPloit	Exploit	OK	OK	OK

c) производится модификация весовых коэффициентов нейрона-победителя в соответствии со следующими выражениями:

$$\omega_{ck}(t+1) = \omega_{ck}(t) + \gamma\,(X_c - \omega_{ck}(t)). \qquad (21)$$

Если при подаче на вход сети чистого фрагмента победителем является один из первых p нейронов или при подаче на вход сети вредоносного фрагмента победителем является один из r последних нейронов сети Кохонена. В противном случае:

$$\omega_{ck}(t+1) = \omega_{ck}(t) - \gamma\,(X_c - \omega_{ck}(t)). \qquad (22)$$

3. Процесс повторяется, начиная с пункта 2 для всех входных образов.

4. Обучение производится до желаемой степени согласования между входными и весовыми векторами, т.е. до тех пор, пока значение суммарной квадратичной ошибки не станет равной нулю (формула 5).

Основное отличие предложенного алгоритма от известных заключается в том, что при корректной классификации каждому входному образу соответствует не конкретный нейрон, а один из первых *p* нейронов, или один из последних *r* нейронов слоя Кохонена.

Третий слой, состоящий из двух линейных нейронных элементов, осуществляет отображение кластеров, сформированных слоем Кохонена, в два класса, которые характеризуют чистые и вирусные входные образы. В общем случае выходное значение *j*-го нейрона третьего слоя определяется следующим образом:

$$Y_j = \sum_{i=1}^{m} \omega_{ij} \cdot Y_i\ , \qquad (23)$$

где ω_{ij} – весовой коэффициент между *i*-м нейроном слоя Кохонена и *j*-м нейроном линейного слоя.

Если нейрон-победитель в слое Кохонена имеет номер k, то выходное значение *j*-го нейрона третьего слоя равняется

$$Y_j = \omega_{kj} \cdot Y_k \qquad (24)$$

Для соответствующего отображения входных образов в два класса матрица весовых коэффициентов третьего слоя должна формироваться следующим образом:

$$\omega_{kj} = \begin{cases} 1, & \text{если } k = 1,2 \ldots p \text{ и } j = 1 \text{ или} \\ k = p+1 \ldots r \text{ и } j = 2 \\ 0, & \text{если } k = 1,2 \ldots p \text{ и } j = 2 \text{ или} \\ k = p+1 \ldots r \text{ и } j = 1. \end{cases} \qquad (25)$$

Так, например, для p=8 и r=2, получается следующая матрица весовых коэффициентов

$$W^T = \begin{bmatrix} 1\,1\,1\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1 \end{bmatrix}. \qquad (26)$$

Результаты экспериментов. В таблице 2 представлены результаты сравнительного анализа обнаружения вредоносных программ различными антивирусными продуктами. Для теста были выбраны следующие антивирусные продукты: Антивирус Касперского [8] с актуальными вирусными базами; Антивирус Касперского с устаревшими вирусными базами; антивирусный продукт NOD 32 [9] с отключенными вирусными базами, но с задействованным эвристическим анализатором и разработанная нами нейросетевая искусственная иммунная система. В таблице «*ОК*» – означает решение антивирусной программы о том, что файл является чистым.

Как видно из полученных результатов, антивирус с актуальными вирусными базами обнаружил все вредоносные программы, которые использовались в эксперименте. Это объясняется тем, что в антивирусных базах содержались сигнатуры используемых в эксперименте вредоносных программ. Антивирус с устаревшими базами обнаружил только половину присутствующих вирусов, что наглядно отражает неспособность сигнатурного метода обнаруживать неизвестные вредоносные программы. Антивирус NOD 32, который использовал эвристический анализатор, обнаружил только семь вирусов, что является очень низким показателем для надежной современной системы безопасности и отражает проблемную ситуацию обнаружения неизвестных вредоносных программ с помощью эвристических методов. Искусственная иммунная система показала наилучшие результаты. Один нейросетевой иммунный детектор способен обнаруживать несколько вредоносных программ. Причем детектор приобретает способность обнаруживать принципиально новые вредоносные программы.

Заключение. Разработанные принципы построения нейросетевой искусственной иммунной системы позволяют конструировать интеллектуальные системы обнаружения вредоносных программ, способных к адаптации и самоорганизации. Разработаная структура нейросетевого иммунного детектора для обнаружения вредоносных программ, которая состоит из трех слоев нейронных элементов и арбитра, характеризуется малым объемом обучающей выборки и отношением количества нейронов в слое Кохонена, характеризующих соответственно чистый и вредоносный классы кратно 4/1. Предложенный нейросетевой иммунный детектор способен обнаруживать как известные, так и новые вредоносные программы. Разработанный алгоритм обучения нейросетевого иммунного детектора позволяет эффективно обучать НИД для обнаружения вредоносных программ. Отличительной особенностью НИД является то, что при корректной классификации каждому образу соответствует не конкретный нейрон слоя Кохонена, а совокупность нейронных элементов. Проведенные эксперименты по тестированию нейросетевой искусственной иммунной системы показали способность нейросетевых иммунных детекторов обнаруживать разнотипные неизвестные вредоносные программы.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Киберпреступность // Центр исследования компьютерной преступности [Электронный ресурс]. – 2007. – Режим доступа: http://www.crime-research.ru/news/05.09.2007/3793/. – Дата доступа: 27.11.2007.
2. Пресс-Центр // Антивирус ВирусБлокАда [Электронный ресурс]. – 2005. – Режим доступа: http://www.anti-virus.by/press/viruses/1485.html. – Дата доступа: 25.08.2007.
3. Касперский, Е. Компьютерное зловредство / Е. Касперский. – СПб.: Питер, 2007. – 208 с.
4. Касперский, К. Записки исследователя компьютерных вирусов / К. Касперский. – СПб.: Питер, 2006. – 316 с.
5. Куприянов, А.И. Основы защиты информации / А.И. Куприянов, А.В. Сахаров. – М.: Академия, 2006. – 256 с.
6. Зайцев, О.В. Rootkits, spyware/adware, keyloggers & backdoors: Обнаружение и защита / О.В. Зайцев. – СПб.: BHV-Санкт-Петербург, 2006. – 304 с.
7. Проактивность как средство борьбы с вирусами // Интернет-безопасность [Электронный ресурс]. – 1996. – Режим доступа: http://www.viruslist.com/ru/analysis?pubid=189544544. – Дата доступа: 15.05.2008.
8. Рассел, С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2005. – 1424 с.
9. Дасгупта, Д. Искусственные иммунные системы и их применение / Д. Дасгупта; под ред. Д. Дасгупта. – М.: Физматлит, 2006. – 344 с.
10. Головко, В.А. Нейронные сети: обучение, организация, применение / В.А. Головко // Нейрокомпьютеры и их применение : учеб. пособие / В.А. Головко. – М., 2001. – 256 с.
11. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.
12. Яхъяева, Г.Э. Нечеткие множества и нейронные сети / Г.Э. Яхъяева. – М.: Бином.ЛЗ, 2008. – 316 с.
13. Безобразов, С.В. Искусственные иммунные системы для защиты информации: применение LVQ сети / С.В. Безобразов // Нейроинформатика-2007: материалы IX Всеросс. науч.-техн. конф., Москва, 23–26 января 2007 г. / Московский инженерно-физический институт (государственный университет). – Москва, 2007. – С. 27–35.
14. Безобразов, С.В. Искусственные иммунные системы для защиты информации: обнаружение и классификация компьютерных вирусов / С.В. Безобразов, В.А. Головко // Научная сессия МИФИ «Нейроинформатика»: материалы Всеросс. науч. конф., МИФИ, Москва, 20-23 янв. 2008. – Москва, 2008. – С. 23–27.
15. Bezobrazov, S. Neural Networks for Artificial Immune Systems: LVQ for Detectors Construction / S. Bezobrazov, V. Golovko // IDAACS'2007: proceedings of the 4 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Dortmund, 2007. – P. 180–184.
16. Bezobrazov, S. Neural networks and artificial immune systems – malware detection tool / S. Bezobrazov, V.Golovko // ICNNAI'2008: proceedings of the 5 International Conference on Neural Networks and Artificial Intelligence, Minsk, 27-30 May 2008. / Brest State University of Informatics and Radioelectronics. – Minsk, 2008. – P. 49–52.

Материал поступил в редакцию 19.11.09

BEZOBRAZOV S.V., GOLOVKO V.A. The Neuronet Immune System for Malware Detection: the Principles of Construction

The principles of the neuronet immune system construction for unknown malicious software detection are proposed. Research results are submitted.

УДК 004.5;621.38

Бутов А.А.

МЕТОД ФОРМИРОВАНИЯ ОДНОСВЯЗНЫХ И МНОГОСВЯЗНЫХ МНОГОУГОЛЬНИКОВ В ЗАДАЧАХ ПРОЕКТИРОВАНИЯ ТОПОЛОГИИ СБИС

Введение. При производстве многих микроэлектронных устройств возникает задача формирования топологических структур на металлизированных фотошаблонах [1]. Эти структуры формируются с помощью специальных генераторов изображений, которые строят топологию на фотошаблоне из наборных элементов в виде прямоугольников различных размеров. Вначале топологические структуры представляются в векторной форме набором простых замкнутых ломаных, однако далее для каждой из этих структур формируется изображение, которое получается путем пошагового экспонирования отдельных областей структуры в виде прямоугольников, объединение которых дает изображение всей структуры с заданной точностью. Для получения изображения прямоугольника на экспонируемой поверхности используется перемещаемая диафрагма прямоугольной формы с регулируемыми размерами и ориентацией.

Автоматизированные системы подготовки топологической информации для микрофотонаборных генераторов изображений должны решать целый ряд достаточно сложных задач логико-комбинаторного характера [2, 3]. В настоящей работе предлагается описание метода решения одной из частных задач, которая связанна с формированием односвязных и многосвязных многоугольников. Необходимость решения этой задачи обусловлена тем, что исходная топологическая информация, которая обычно задается множеством простых ломаных (контуров), не может быть непосредственно использована для решения задач покрытия элементов топологии СБИС прямоугольниками [4, 5], так как оказывается невозможным распознать, описывает ли очередной контур часть поверхности, подлежащей покрытию, или это описание той внутренней части поверхности, которую не нужно покрывать.

Основные определения, постановка задачи. Точки плоскости a и b, заданные соответственно координатами (x_a, y_a) и (x_b, y_b) в декартовой системе, где x и y – переменные, связанные соответственно с осью абсцисс OX и с осью ординат OY, совпадают, если $x_a = x_b$ и $y_a = y_b$. Если хотя бы одно из этих равенств не выполняется, то точки считаются различными. *Отрезком* ab называется пара различных точек a и b плоскости, соединенных прямой линией. Точки a и b отрезка ab называются *граничными*. Рассмотрим различные точки плоскости $a, b, c, d, \ldots, m, n$. Соединим эти точки отрезками $ab, bc, cd, \ldots, mn, na$. Получим замкнутую ломаную, которую обозначим через $L = abcd\ldots mn$. Точки $a, b, c, d, \ldots, m, n$ называются *вершинами* ломаной L, а отрезки $ab, bc, cd, \ldots, mn, na$ – ее *сторонами*.

Два отрезка пересекаются, если существует хотя бы одна точка плоскости, принадлежащая каждому из них. Если такая точка отсутствует, то отрезки не пересекаются.

Замкнутая ломаная L является *простой*, если любая точка, общая для двух ее сторон, является граничной для этих и только для этих сторон. В дальнейшем простую ломаную будем называть *контуром*.

Два контура L_1 и L_2 не пересекаются, если ни одна из сторон контура L_1 не пересекается ни с одной из сторон контура L_2.

Рассмотрим некоторый контур L. Этот контур делит плоскость на

Бутов А.А., *аспирант Объединенного института проблем информатики НАН Беларуси.*
Беларусь, ОИПИ НАН Беларуси, 220012, г. Минск, ул. Сурганова, 6.