УДК 004.822:514

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ АНОМАЛИЙ В КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЕ ПРЕДПРИЯТИЯ

В.А. ВИШНЯКОВ, О.И. КОВАЛЬ, М.Г. МОЗДУРАНИ ШИРАЗ

Белорусский государственный университет информатики и радиоэлектроники
П.Бровки, 6, Минск, 220600, Беларусь

Поступила в редакцию 11 марта 2016

Рассмотрены нейросетевые структуры, используемые для решения задачи защиты информации. Построена выборка атрибутов и метаданных исполняемых файлов, которая использовалась для обучения многослойного персептрона. Обучение проводилось в программе SPSS Statistics. После обучения нейронной сети эффективность ее работы была определена с помощью контрольной выборки исполняемых файлов. Относительная погрешность классификации файлов составила 5 %.

Ключевые слова: нейросетевые структуры, защита информации, обучение нейронной сети, корпоративные информационные системы.

Введение

Нейронные сети (НС) могут применяться в компонентах систем оповещения об атаках (СОА), а также в системах обнаружения уязвимостей (СОУ). С помощью многослойного персептрона решается задача распознавания реализации атаки на компьютерную систему. СОА на основании НС получили распространение, однако они обладают рядом существенных недостатков, которые ограничивают их практическую ценность [1]. К указанным недостаткам относятся: высокий уровень ложных тревог, сложность подбора оптимальных граничных параметров, сложность ввода в систему нового субъекта/объекта наблюдений, недостаточная адаптация ко многим особенностям современного состояния отрасли информационных технологий. Это свидетельствует о необходимости дальнейшего усовершенствования таких систем. При этом следует учитывать определенный прогресс в развитии теории НС, что должно отражаться в методике их использования в задачах защиты информации (ЗИ) [2].

Основная сложность в использовании НС заключается в корректном построении такого пространства признаков, которое позволило бы разделить классы атак между собой и отделить их от нормального поведения. Кроме того, для классических НС характерно долгое обучение, при этом время обучения зависит от размера обучающей выборки [3]. В соответствии с введенными критериями, НС используются на сетевом и узловом уровнях, являются адаптивными, имеют сравнительно низкую вычислительную сложность. При этом они не являются верифицируемыми и устойчивы, как правило, только в пределах той сети, в которой они обучались, что существенно ограничивает применимость метода.

Нейронные сети для обнаружения аномалий обучаются в течение некоторого периода времени, когда все наблюдаемое поведение считается нормальным. После обучения нейронная сеть запускается в режиме распознавания [1]. В ситуации, когда во входном потоке не удается распознать нормальное поведение, фиксируется факт атаки. В случае использования репрезентативной обучающей выборки НС дают хорошую устойчивость в пределах заданной системы; но составление подобной выборки является серьезной и сложной задачей.

Методика эксперимента

Построим обучающую выборку для многослойного персептрона, определяющего, «заражена» ли данная программа (ее исполняемый файл) «вирусом» или нет. Входными параметрами НС будут служить различные атрибуты и метаданные, извлеченные из PE-структуры (Portable Executable structure) исполняемого файла, так как они предоставляют достаточно много информации о коде внутри исполняемых файлов и которые могут помочь определить, является ли файл зараженным (любой зараженный файл имеет похожие значения атрибутов PE-структуры) [4]. Преимуществом такого подхода является то, что необходимая для обучения НС информация может быть получена без запуска исполняемого файла, что минимизирует риск заражения компьютерной системы вирусом.

PE-формат (Portable Executable) − это формат всех 32- и 64-разрядных исполняемых файлов в ОС Windows. Такой формат имеют файлы EXE, DLL, SYS, DRV, MSSTYLE, MUI, CPL, OCX, BPL, DPL, SCR [5]. Каждый исполняемый файл формата PE состоит из множества взаимосвязанных структур, содержащих информацию о самом файле, об импортируемых и экспортируемых им функциях, о перемещаемых элементах, ресурсах и т. д. (рис. 1).

Каждый PE-файл состоит из вышеперечисленных элементов, они являются обязательными. Любой PE-файл состоит из нескольких заголовков и нескольких (от 1 до 96) секций [4]. Заголовки содержат служебную информацию, описывающую различные свойства исполняемого файла и его структуру. Секции содержат данные, которые размещаются в адресном пространстве процесса во время загрузки исполняемого файла в память [5].

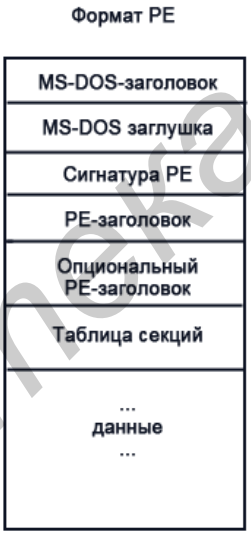

Рис. 1. Структура PE-файла

Сверху находится MS-DOS заголовок и MS-DOS заглушка – наследие времен, когда происходил переход с DOS на Windows, поддерживающий новый PE-формат. Далее идет сигнатура PE-файла (4 байта: 'P', 'E', 0, 0), после которой начинается структура IMAGE_FILE_HEADER, содержащая в себе следующие поля [5]:

1. Machine − архитектура, на которой может запускаться файл;
2. NumberOfSections − количество секций в PE-файле. Допустимое значение − от 1 до 0x60. Секция − это некая область памяти, обладающая определенными характеристиками и выделяемая системой при загрузке исполняемого файла;
3. SizeOfOptionalHeader − размер структуры опционального заголовка в байтах;
4. Cracteristics − поле флагов характеристик PE-файла. Тут содержится информация о том, имеет ли файл экспортируемые функции, перемещаемые элементы, отладочную информацию. Остальные поля при загрузке ни на что не влияют.

Далее идет опциональный заголовок PE-файла. На самом деле, никакой он не опциональный, без него файл загружен не будет, хотя размер этого заголовка может варьироваться. Его основные поля [5]:

1. Magic − для 32-разрядных PE-файлов это поле должно содержать значение 0x10B, а для 64-разрядных − 0x20B;

2. AddressOfEntryPoint − адрес точки входа относительно базового адреса загрузки файла (ImageBase);

3. ImageBase − базовый адрес загрузки PE-файла. Если у файла имеется таблица перемещаемых элементов, то этот адрес может варьироваться, а ImageBase будет содержать лишь рекомендуемый адрес загрузки;

4. FileAlignment и SectionAlignment − файловое и виртуальное выравнивание секций. В обязательном порядке должны быть выполнены следующие условия: SectionAlignment >= 0x1000; FileAlignment >= 0x200; SectionAlignment >= FileAlignment.

5. SizeOfImage − это поле содержит размер в байтах загруженного образа PE-файла, который должен быть равен виртуальному адресу последней секции плюс ее виртуальный выровненный размер.

6. SizeOfHeaders − размер всех заголовков. Это поле говорит загрузчику, сколько байт считать от начала файла, чтобы получить всю необходимую информацию для загрузки файла. Значение поля не должно превышать относительного виртуального адреса первой секции.

7. CheckSum − контрольная сумма файла, которая проверяется загрузчиком только для самых важных системных файлов.

8. Subsystem − подсистема файла. Самые распространенные − IMAGE_SUBSYSTEM_WINDOWS_GUI (GUI-интерфейс Windows) и IMAGE_SUBSYSTEM_WINDOWS_CUI (консольный интерфейс).

9. SizeOfStackReserve и SizeOfStackCommit, SizeOfHeapReserve и SizeOfHeapCommit − размер соответственно стека и кучи, которые должны быть зарезервированы и выделены для PE-файла. 0 − значение по умолчанию; в случае, если SizeOfStackCommit > SizeOfStackReserve или SizeOfHeapCommit > SizeOfHeapReserve, то файл загружен не будет.

Для извлечения атрибутов и метаданных из PE-структуры исполняемых файлов используется программа PE Explorer, которую можно скачать на официальном сайте [6]. Исполняемые файлы, как «чистые», так и зараженные вирусами, были предоставлены отделами информационной безопасности СП ЗАО «IBA Group». Обучающая выборка была составлена из метаданных 100 файлов. Все файлы поочередно загружались в PE Explorer, откуда их атрибуты и метаданные в шестнадцатеричном представлении сохранялись в Excel-таблицу.

Сравнительный анализ атрибутов и метаданных чистых и зараженных вирусами файлов показал, что значения некоторых полей, составляющих структуру PE-файла, представленную на рис. 1, являются одинаковыми для обоих типов файлов. Такими полями являются: PointerToSymbolTable, NumberOfSymbols, SizeOfOptionalHeader, Magic, BaseOfCode, ImageBase, SectionAlignment, Win32VersionValue, Subsystem, SizeOfStackReverse, SizeOfHeapCommit, LoaderFlags, NumberOfDataDirectories. Это позволяет не использовать значения вышеперечисленных полей, так как они не несут никакой информации о безопасности исполняемого файла и никак не повлияют на дальнейшее обучение нейронной сети.

Для обучения многослойного персептрона данные обучающей выборки должны быть приведены к десятичному представлению. Для автоматизации решения данной задачи на языке JavaScript был написан конвертер чисел между различными системами счисления. После перевода шестнадцатеричных данных в десятичную систему счисления обучающая выборка принимает данные.

Итоговая выборка состоит из таких полей, как NumberOfSections, TimeDateStamp, Characteristics, LinkerVersion, SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, AdressOfEntryPoint, BaseOfData, FileAlignment, OperatingSystemVersion, ImageVersion, SubsystemVersion, SizeOfImage, SizeOfHeaders, Checksum, DllCharacteristics, SizeOfStackCommit. Обучение сети проводится с использованием SPSS Statistics – ПО компании IBM и предназначенного для статистической обработки информации и данных, а также являющегося лидером среди программных продуктов для статистического анализа.

В SPSS Statistics предусмотрены два вида нейронной сети: многослойный персептрон и радиальная базисная функция. Многослойный персептрон в SPSS Statistics может содержать одну или более зависимых величин, значения которых определяются как функции независимых величин. Зависимые величины могут быть: номинальные, или категориальные; ординальные, или порядковые; численные. Независимые величины могут быть как категориальными, так и

численными. В нашем случае обучения НС для определения безопасности исполняемых файлов все зависимые и независимые величины будут численными.

Многослойный персептрон в SPSS Statistics может иметь один или два скрытых слоя. Количество нейронов в скрытых слоях может быть определено автоматически программой или установлено вручную. Активационная функция может быть четырех видов: единичная активационная функция $\gamma(c) = c$; логистическая функция $\gamma(c_k) = \frac{e^{c_k}}{\sum_{c_j} c_j}$, применяется для категориальных величин; гиперболический тангенс $\gamma(c) = \text{th(c)} = \frac{(e^c - e^{-c})}{(e^c + e^{-c})}$, преобразует реальные значения аргументов к диапазону $(-1,1)$; сигмоида $\gamma(c) = \frac{1}{(1+e^{-c})}$, преобразует значения аргументов к диапазону $(0,1)$.

В обучаемой НС будет 18 входных нейронов и 1 выходной. SPSS Statistics предлагает два варианта обучения многослойного персептрона: с одним либо с двумя скрытыми слоями, для решения задачи защиты информации будет обучать персептрон с двумя скрытыми слоями, в первом из которых будет три нейрона, а во втором – два.

В SPSS Statistics предусмотрены три режима обучения многослойного персептрона.

1. Batch-режим. Подстраивает синаптические веса только после загрузки всех обучающих примеров (использует информацию из всех записей в обучающей выборке одновременно). Этот режим предпочтителен, так как он минимизирует общую ошибку, но этот режим необходимо применять неоднократно до тех пор, пока одно из останавливающих обучение правил не будет достигнуто при обучении сети. Batch-режим наиболее эффективен для малых выборок.

2. Online-режим. Подстраивает синаптические веса после каждой отдельной записи обучающего примера. Используется для больших по размеру выборок, чем выборки в Batch-режиме; если входных параметров достаточно много, и их значения не зависят друг от друга, то в этом случае наиболее эффективен Online-режим.

3. Mini-batch-режим. Разбивает обучающую выборку на группы приблизительно одинакового размера, после чего подстраивает веса после «прогона» каждой группы обучающих примеров по НС. Данный режим обучения отлично подходит для выборок, средних по величине.

При обучении многослойного персептрона для решения задачи ЗИ используется Batch-режим, так как выборка элементов невелика и составляет 100 исполняемых файлов.

Результаты и их обсуждение

Итоговая конфигурация многослойного персептрона для определения состояний исполняемых файлов выглядит следующим образом (рис. 2). Величина, называемая смещением позволяет управлять уровнем активации нейрона. Сдвигая активационную функцию вправо или влево вдоль горизонтальной оси, увеличивая смещение, повышаем порог активации и искусственно вводим некоторое торможение нейрона, а уменьшая, как бы «подталкиваем» нейрон (табл. 1).

Распределение исполняемых файлов обучающей выборки: 74 элемента составили обучающую выборку, а 21 был использованы в качестве тестовой выборки с целью предварительного определения точности классификации НС. Сначала НС обучалась на 74 элементах из первой выборки, затем на ее входы подавались значения полей исполняемых файлов из тестовой выборки, и НС определяла их состояния и сравнивала с реальными, в случае отличия от реального значения проходило ее дообучение. Контрольная выборка, состоящая из 5 элементов, используется для итогового тестирования НС.

Табл. 1 отражает информацию о конфигурации НС, а также о типе используемых активационной функции и функции ошибок.

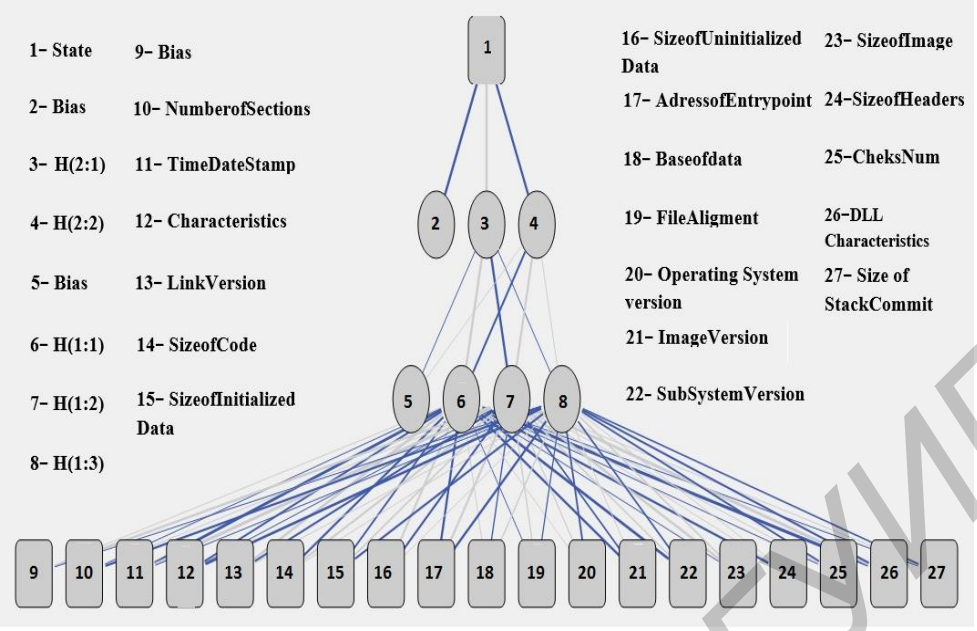

Рис. 2. Конфигурация многослойного персептрона

Таблица 1. **Конфигурация нейронной сети, типы активацинной функции и функции ошибок**

Network Information			
Input Layer	Covariates	1	Number of sections
		2	Time Date Stamp
		3	Characteristics
		4	Linker Version
		5	Size of Code
		6	Size of Initialized data
		7	Size of Uninitialized data
		8	Adress of Entry Point
		9	Base of Data
		10	File Alignment
		11	Operating System Version
		12	Image Version
		13	Subsystem Version
		14	Size of Image
		15	Size of Headers
		16	Checksum
		17	Dll Characteristics
		18	Size of Stack Commit
		Number of Units	18
		Rescaling Method for Covariates	Standardized
Hidden Layer(s)		Number of Hidden Layers	2
		Number of Units in Hidden Layer 1	3
		Number of Units in Hidden Layer 2	2
		Activation Function	Hyperbolic tangent
Output Layer	Dependent Variables	1	STATE
	Number of Units		1
	Rescaling Method for Scale Dependents		Adjusted normalized
	Activation Function		Hyperbolic tangent
	Error Function		Sum of Squares

Таблица 2**. Описание результатов модели**

Model Summary		
Training	Sum of Squares Error	2,429
	Relative Error	0,071
	Stopping Rule Used	1 consecutive step(s) with no decrease in error[a]
	Training Time	0:00:00.047
Testing	Sum of Squares Error	2,577
	Relative Error	0,266
Holdout	Relative Error	0,001
Dependent Variable: STATE		
Error computations are based on the testing sample		

Табл. 2 отображает результаты обучения и применения НС к контрольной выборке исполняемых файлов.

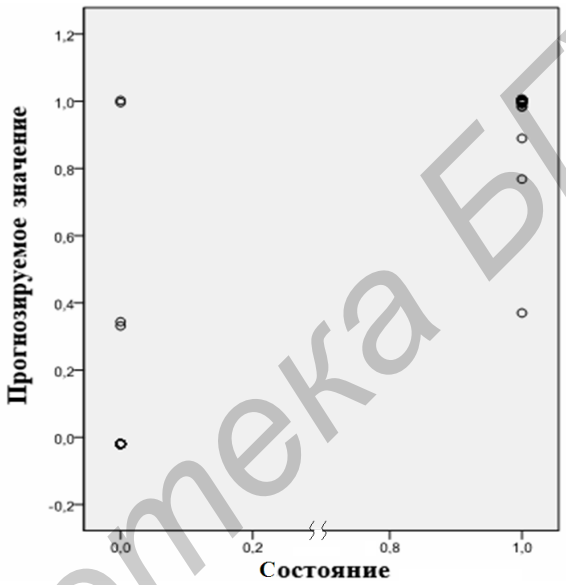

Рис. 3. Результаты предсказания состояний исполняемых файлов из тестовой выборки на этапе дообучения НС

Так как в контрольную выборку были объединены 5 исполняемых файлов с неизвестными нам состояниями, то было проведено дополнительное тестирование обученной нейронной сети на выборке элементов, состоящей из 10 чистых исполняемых файлов и 10 файлов, зараженных вирусами. В результате данного тестирования все чистые файлы были верно классифицированы, 9 зараженных файлов были отнесены к категории вирусов, а один отнесен к категории чистых. Таким образом, погрешность классификации исполняемых файлов составила 5 %, но следует отметить, что при обучении НС использовалась относительно небольшая выборка исполняемых файлов. Для использования НС при решении реальных задач защиты информации, в больших внутрикорпоративных системах, выборка файлов должна быть как можно больше, и вирусы, которыми часть файлов из выборки заражена, должны быть как можно более разнообразными.

Заключение

Рассмотрены основные нейросетевые структуры, используемые для решения задачи ЗИ. Построена выборка атрибутов и метаданных исполняемых файлов двух состояний: чистых и зараженных вирусом. Данная выборка использовалась для обучения многослойного персептрона – наиболее распространенной нейросетевой структуры. Обучение проводилось в SPSS Statistics – программе, произведенной компанией IBM и предназначенной для статистической обработки информации и данных.

После обучения НС эффективность ее работы была определена с помощью контрольной выборки исполняемых файлов. Относительная погрешность классификации файлов составила 5 %, что является достаточно хорошим результатом. Следует, однако, отметить, что при обучении НС использовалась относительно небольшая выборка исполняемых файлов; для использования же НС при решении реальных задач ЗИ для больших КИС, выборка файлов должна быть как можно больше, и вирусы, которыми часть файлов из выборки заражена, – разных видов.

USE OF NEURAL NETWORKS FOR DETECTION AND RECOGNITION OF THE ANOMALIES IN ENTERPRISE CORPORATIVE INFORMATION SYSTEM

U.A. VISHNIAKOU, O.S. KOVAL, M.G. MOZDURANI SHIRAZ

Abstract

The neural networks structure using for task solving of information defense are discussed. The choice of attribute and metadata of executing files with two states (clean and with viruses) which used for multilevel perseptron teaching are built. The teaching was realized within SPSS Statistics – program of IBM Company. After the teaching of neural network the efficiently its working with the control choice of executing files was determined. The relative value of files classification was 5 %, that it is the good result. The file choice must be greater and viruses more variables within the use such approach for large enterprise corporative information systems.

Key words: neural network structure, information defense, viruses file, teaching of neuron net, corporative information system.

Список литературы

1. *Головко В.А.* Нейронные сети: обучение, организация и применение. М., 2001.
2. *Вишняков В.А.* Информационное управление и безопасность: методы, модели, программно-аппаратные решения. Минск, 2014.
3. *Bishop C.M.* Neural Networks for Pattern Recognition. Oxford, 1955.
4. *Shivani Shah, Himali Jani, Sathvik Shetty et.al.* // International Journal of Computer Applications. 2013. Vol. 84, № 5. P. 17–23.
5. Matt Pietrek «Peering Inside the PE: A Tour of the Win32 Portable Executable File Format» [Электронный ресурс]. − Режим доступа : https://msdn.microsoft.com/en-us/library/ms809762.aspx. Дата доступа: 10.12.2015.
6. Программа PE Explorer [Электронный ресурс]. − Режим доступа: http://www.pe-explorer.com/. Дата доступа: 10.12.2015.