единственный размерный параметр, алгоритм имеет меньшую вычислительную сложность по сравнению с алгоритмами, в которых используются иные фигуры – например, эллипсы.

Процедура изометрического покрытия достаточно устойчива к операции предварительной фильтрации изображения. Кроме того, она предусматривает настройку параметров в виде задания допустимых пределов для величины перекрытия дисков и области твердой фазы, а также для размера непокрытых фрагментов поровой области.

Алгоритм позволяет получать как интегральную характеристику пористости материала – полную пористость, так и дифференциальную функцию распределения площади (либо количества) пор по размеру. Получаемые эмпирические функции соответствуют типовым характеристикам пористости цементных композитов, получаемых иными, в том числе физическими, методами. На основании полученных данных, с применением интегрального реконструктивного преобразования, могут быть определены также функции дифференциальной пористости для объема материала.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Шейкин А.Е., Чеховский Ю.В., Бруссер М.И. Структура и свойства цементных бетонов. – М.: Стройиздат, 1979. – 344 с.
2. Пантелеев В.Г., Егорова О.В., Клыкова Е.И. Компьютерная микроскопия. – М.: Техносфера, 2005. – 304 с.
3. Vočka R., Gallé Ch., Dubois M., Lovera P. Mercury intrusion porosimetry and hierarchical structure of cement pastes. Theory and experiment // Cement & Concrete Research. – 2000, Vol. 30. – P. 521-527.
4. Ye G., van Breugel K., Fraaij A.L.A. Three-dimensional microstructure analysis of numerically simulated cementious materials // Cement & Concrete Research. – 2003, Vol. 33. – P. 215-222.
5. Hu J., Stroeven P. Depercolation threshold of porosity in model cement: approach by morphological evolution during hydration // Cement & Concrete Composites. – 2005, Vol. 27. P. 19-25.
6. Медведев Н.Н. Метод Вороного-Делоне в исследовании структуры некристаллических систем / РАН, Сиб. отдние, РФФИ, Ин-т химической кинетики и горения СО РАН. – Новосибирск: НИЦ ОИГГМ СО РАН, Издательство СО РАН, 2000. – 214 с.
7. Прэтт У. Цифровая обработка изображений. – М.: Мир, 1982. – Кн. 2. – 480 с.
8. Дереченник С.С., Разумейчик В.С., Тур В.В. Закономерности топологической неупорядоченности в плоских сечениях и объемах дисперсных систем // Вестник БГТУ. Сер. Строительство и архитектура. – 2005. – № 2 (32). – С. 18-25.

Статья поступила в редакцию 28.01.2007

УДК 004.8.032.26

Головко В.А., Войцехович Л.Ю., Шевеленков В.В.

НЕЙРОСЕТЕВЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ НЕЙРОННЫХ СИСТЕМ ОБНАРУЖЕНИЯ АТАК НА КОМПЬЮТЕРНЫЕ СЕТИ

1. ***Введение***

Одной из форм глобализации мирового пространства является информационная глобализация, которая связана с повсеместным распространением сети Интернет. Определяющим направлением развития компьютерной отрасли стало внедрение и расширение сетевых систем.

В результате этого значительно возросло количество атак и злоупотреблений в сфере высоких технологий. Поэтому вопросу безопасности компьютерных систем уделяется все больше и больше внимания.

Задачей Систем Обнаружения Атак (Intrusion Detection Systems - IDS) является защита компьютерных сетей. В последнее время системы IDS активно изучаются. Они должны выполнять свои функции в режиме реального времени. Существует два основных метода в сфере обнаружения атак: обнаружение злоупотреблений (misuse detection) и обнаружение аномалий (anomaly detection). Обнаружение злоупотреблений предполагает наличие сигнатур атак. Основным недостатком таких систем является их неспособность обнаруживать новые или неизвестные атаки, т.е. записи о которых в системе отсутствуют. Примерами систем обнаружения злоупотреблений могут служить: IDIOT [1], STAT [2] и Snort [3]. Обнаружение аномалий связано с построением профиля нормального поведения пользователя. Причем атакой считается любое отклонение от этого профиля. Главным преимуществом таких систем является принципиальная возможность определения ранее не встречавшихся атак. Примеры таких систем: IDES [4] и EMERALD [5].

В настоящее время разрабатывается большое количество различных технологий защиты компьютерных сетей, которые базируются на применении нейронных сетей (neural networks), на технологиях извлечения данных (data mining), статистическом анализе и т.п. Так, например, классификатор главных компонент представлен в работах [6, 7]. Различные технологии извлечения данных описаны в [8, 9]. Другие авторы предлагают геометрические структуры в задачах обнаружения атак, это: алгоритмы кластеризации, алгоритмы k-Nearest Neighbor (k-NN) и Support Vector Machine (SVM) [10, 11]. Кроме того, для обнаружения атак могут применяться различные нейронные сети [12, 13]: самоорганизующиеся карты Кохонена (Self Organizing Maps - SOM), многослойный пересептрон (Multilayer Perceptron - MLP), сети радиальнобазисной функции (Radial Basis Function - RBF).

К недостаткам существующих моделей IDS, в первую очередь, можно отнести уязвимость к новым атакам, низкая точность и скорость работы. Современные системы обнаружения вторжений плохо приспособлены к работе в реальном режиме времени, в то время как возможность обрабатывать большой объем данных в реальном режиме времени – это определяющий фактор практического использования систем IDS.

В нашей предыдущей статье [14] рассматривались различные варианты архитектур систем IDS, которые были основаны на применении комбинаций линейной рециркуляционной нейронной сети (Recirculation Neural Network - RNN) и многослойного персептрона. Эта статья является продолжением предыдущих работ, и здесь предлагаются новые модели: линейная RNN (LRNN) и MLP, нелинейная RNN (NRNN) и MLP, Ансамблевая нейронная сеть (Ensembling Network - EN). Задачей RNN является сжатие входного пространства образов с целью получения главных компонент. Многослойный персептрон производит основные вычисления, связанные с распознаванием входного вектора, используя информацию, предоставленную рециркуляционными нейронными сетями.

Головко Владимир Адамович, *д.т.н., заведующий кафедрой интеллектуальных информационных технологий БрГТУ.*
Войцехович Леонид Юрьевич, *магистрант кафедры интеллектуальных информационных технологий БрГТУ.*
Шевеленков Виталий Вячеславович, *ассистент кафедры интеллектуальных информационных технологий БрГТУ.*
Беларусь, Брестский государственный технический университет, 224017, Беларусь, г. Брест, ул. Московская, 267.

Рис. 1. *Процесс обнаружения*

Статья организована следующим образом. Основные фазы процесса обнаружения и данные, использованные в работе, описаны в разделе 2. В разделе 3 представлены нейросетевые системы обнаружения атак, основанные на модулярной структуре. В разделе 4 рассматриваются линейные и нелинейные рециркуляционные сети (RNN). В разделе 5 описываются ансамблевые и MLP нейронные сети, а также правила их обучения. Результаты экспериментов приведены в разделе 6. Выводы сделаны в последнем разделе.

2. Процесс обработки информации в IDS

Процесс обработки информации в IDS приведен на рис. 1. Он включает три этапа.

На первом этапе осуществляется захват трафика сети (feature selection). Сбор необходимых данных выполняет специальное программное средство (sniffer). В этой работе мы использовали базу данных KDD-99 [15]. Эта база содержит около 5 000 000 записей о соединениях. Каждая запись в этой базе представляет собой образ сетевого соединения. Соединение – последовательность TCP пакетов за некоторое конечное время, моменты начала и завершения которого четко определены, в течение которого данные передаются от IP-адреса источника на IP-адрес приемника (и в обратном направлении) используя некоторый определенный протокол.

Каждая запись о соединении включает 41 параметр сетевого трафика и промаркирована как “атака” или “не атака”. Отдельная запись состоит из около 100 байт. 34 параметра о соединении – числовые, а 7 представлены символьными последовательностями. Например, первый параметр определяет длительность соединения, второй – указывает используемый протокол, третий – целевую службу и т.д. Тем не менее, на первом этапе все параметры конвертируются в числовое представление.

Второй этап связан с уменьшением размерности входного вектора данных и получением главных компонент (feature extraction). Между используемыми параметрами существуют сложные взаимосвязи, которые достаточно тяжело проследить. Некоторые данные являются избыточными. Большое количество параметров может значительно увеличить время вычислений, поэтому этап получения главных компонент является важным этапом в процессе функционирования предлагаемых IDS. В этой статье рассматриваются линейные и нелинейные RNN, выполняющие эти функции. В результате экспериментов было определено оптимальное число главных компонент – 12.

Третий этап состоит в обнаружении и распознавании атак (classification). В базе KDD-99 представлены 22 типа атаки. При этом атаки делятся на четыре основные категории: DoS, U2R, R2L и Probe.

Атака DoS – отказ в обслуживании, характеризуется генерацией большого объема трафика, что приводит к перегрузке и блокированию сервера.

Атака U2R предполагает получение зарегистрированным пользователем привилегий локального суперпользователя (администратора).

Атака R2L характеризуется получением доступа незарегистрированного пользователя к компьютеру со стороны удаленной машины.

Атака Probe заключается в сканировании портов с целью получения конфиденциальной информации.

Каждый класс в свою очередь состоит из отдельных типов атак.

3. Архитектурные решения IDS

Рассмотрим различные архитектурные решения для построения систем обнаружения атак. Они основаны на применении модулярных нейронных сетей. В качестве входных данных используется 41-размерный вектор, который характеризует параметры соединения сети. Задачей IDS является обнаружение и распознавание атак. Поэтому в качестве выходных данных используется 5-мерный вектор, где 5 - это количество классов атак плюс нормальное состояние. Возникает резонный вопрос по структуре IDS: какие параметры входного вектора наиболее значимы для успешного обнаружения того или иного типа атаки? Мы предлагаем использовать рециркуляционную нейронную сеть (RNN) для получения главных компонент.

Третий этап функционирования IDS связан с обнаружением и распознаванием атак. Для этих целей в работе предполагается использование многослойного персептрона (MLP). Комбинируя RNN и MLP нейронные сети, мы можем получать различные архитектуры систем обнаружения атак.

На основании предыдущих результатов экспериментов мы отобрали три наиболее удачных модели систем обнаружения атак.

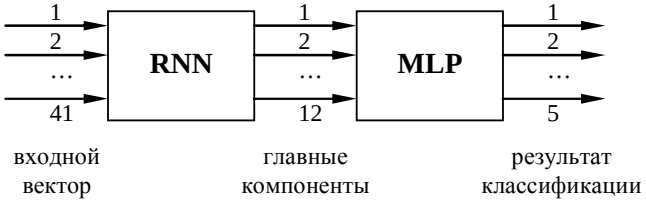

Рис. 2. Первый вариант IDS.

На рис. 2 приведена система обнаружения атак, которая состоит из рециркуляционной нейронной сети и многослойного персептрона, которые соединены последовательно. Задачей RNN является сжатие входного 41-размерного вектора в 12-размерный выходной вектор. Многослойный персептрон осуществляет обработку сжатого пространства входных образов (главных компонент) с целью распознавания класса атаки.

На рис. 3 приведена вторая схема системы обнаружения атак. Она характеризуется тем, что главные компоненты с выходов RNN одновременно поступают на 4 отдельных мно-

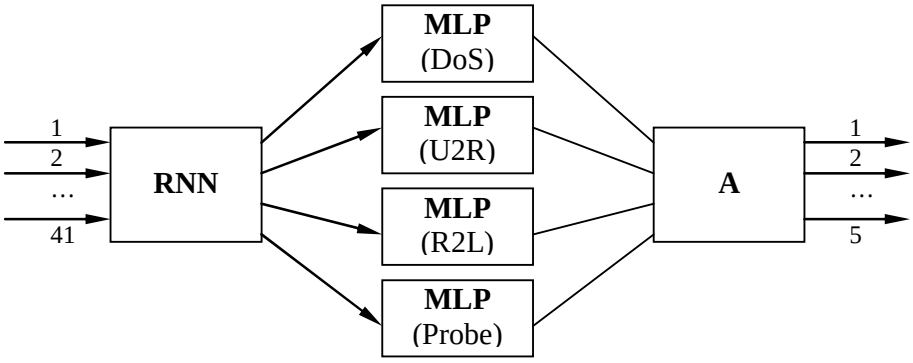

Рис. 3. *Второй вариант IDS*

гослойных персептрона, каждый из которых соответствует определенному классу атаки: DoS, U2R, R2L и Probe. С выходов MLP данные поступают на арбитр, который и принимает окончательное решение о состоянии системы. В качестве арбитра может использоваться линейный или многослойный персептрон. Тогда обучение его будет производиться после обучения RNN и MLP. Такая схема может осуществлять иерархическую классификацию атак. В этом случае арбитр определяет один из 5 классов атаки, а соответствующий многослойный персептрон – тип атаки.

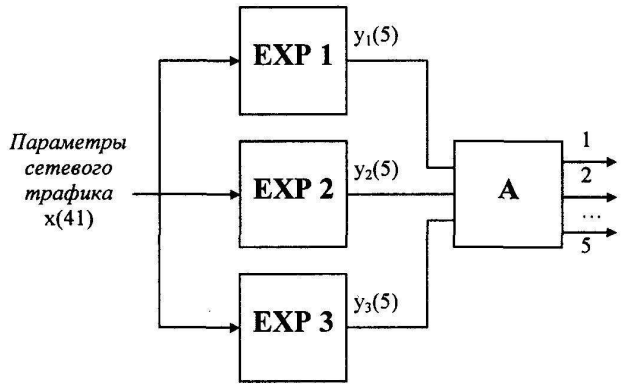

Рис. 4. *Четвертый вариант IDS (режим тестирования)*

Сложные вычислительные задачи решаются при помощи их разбиения на множество небольших и простых задач с последующим объединением полученных решений. Вычислительная простота достигается за счет распределения задачи обучения среди множества экспертов. Комбинацию таких экспертов (EXP) называют Ассоциативной машиной (Committee Machine). По сути, она интегрирует знания, накопленные экспертами, в общее решение, которое имеет приоритет над каждым решением отдельного эксперта.

Следующий вариант структуры IDS основан на этой идее (рис. 4). Каждый эксперт представляет собой отдельную систему классификации. В качестве эксперта мы использовали модель 1. Обучение каждого эксперта происходит на отдельном множестве данных, т.е. данные для обучения каждого последующего эксперта формируются с учетом результатов обучения предыдущих экспертов. Алгоритм, используемый для такого обучения, называют алгоритмом усиления за счет фильтрации (boosting by filtering) [16]. После обучения нейронные сети способны обнаруживать атаки. В режиме тестирования на вход каждого эксперта подается исходный 41-размерный вектор. Арбитр принимает окончательное решение.

4. Рециркуляционная нейронная сеть

В этом разделе рассматриваются две нейронные сети, предназначенные для формирования главных компонент: линейная и нелинейная рециркуляционные сети.

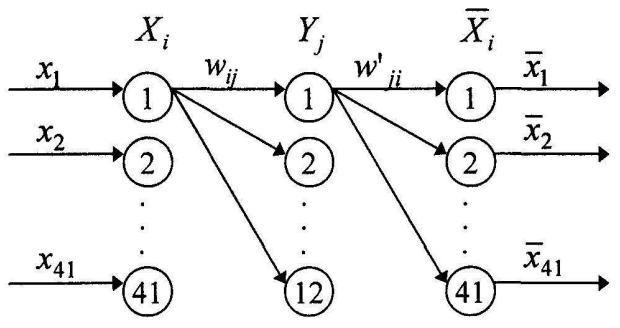

Рис. 5. *Архитектура RNN*

Рассмотрим рециркуляционную нейронную сеть (рис. 5). Она представляется многослойным персептроном, который осуществляет линейное или нелинейное сжатие входных данных через “узкое горлышко” в скрытом слое. Как видно, сеть состоит из трех слоев. Скрытый слой осуществляет сжатие входных образов. Значение j-го элемента скрытого слоя определяется по формулам:

$$y_j = F(S_j),$$

$$S_j = \sum_{i=1}^{41} w_{ij} \cdot x_i,$$

где F – функция активации; S_j – взвешенная сумма j-го нейрона; w_{ij} – весовой коэффициент между i-ым нейроном и j-ым нейроном скрытого слоя; x_i – i-ый входной элемент.

Значение выходных элементов определяется следующим образом:

$$x_i = F(S_i),$$

$$S_i = \sum_{i=1}^{12} w'_{ji} \cdot y_j.$$

В этой статье мы исследуем два алгоритма обучения RNN. Первый алгоритм – это линейное правило обучения Ойя, второй – алгоритм обратного распространения ошибки для нелинейной рециркуляционной нейронной сети.

Весовые коэффициенты линейной RNN модифицируются в соответствии с правилом Ойя [17]:

$$w'_{ji}(t+1) = w'_{ji}(t) - \alpha \cdot y_j \cdot (\overline{x_i} - x_i),$$

$$w_{ij} = w'_{ji}.$$

Как известно, такая RNN осуществляет операцию линейного сжатия. В результате такого преобразования компоненты выходного вектора являются некоррелированными между собой, и первые главные компоненты содержат наиболее информативную составляющую входных данных.

Перед подачей данных на вход RNN проводилась их предварительная обработка:

$$x_i^k = \frac{x_i^k - \mu(x_i)}{\sigma(x_i^k)},$$

где $\mu(x_i) = \frac{1}{L}\sum_{k=1}^{L} x_i^k$,

$$\sigma(x_i^k) = \frac{1}{L}\sum_{k=1}^{L} (x_i^k - \mu(x_i))^2.$$

Здесь L – размерность обучающей выборки.

Как уже упоминалось ранее, для обучения нелинейной RNN используется алгоритм обратного распространения ошибки. Весовые коэффициенты пересчитываются по формулам:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \cdot \gamma_j \cdot F'(S_j) \cdot x_i,$$

$$w'_{ji}(t+1) = w'_{ji}(t) - \alpha \cdot (\overline{x_i} - x_i) \cdot F'(S_i) \cdot y_j,$$

где γ_j - ошибка j-го нейрона,

$$\gamma_j = \sum_{i=1}^{41} (\overline{x_i} - x_i) \cdot F'(S_i) \cdot w'_{ji}.$$

В процессе обучения весовые коэффициенты скрытого слоя ортонормируются в соответствии с процедурой Грамма-Шмидта:

А) В качестве первого вектора ортонормированного базиса выбираем

$$w'_1 = [\frac{w_{11}}{|w_1|}, \frac{w_{21}}{|w_1|}, ..., \frac{w_{n1}}{|w_1|}],$$

где

$$|w_1| = \sqrt{w_{11}^2 + w_{21}^2 + ... + w_{n1}^2}\ .$$

Б) Остальные весовые векторы определяются рекурсивным образом в соответствии со следующими выражениями:

$$w_i = w_i - \sum_{j=1}^{i-1} (w_i^T \cdot w'_j) \cdot w'_j\ ,$$

$$w'_j = [\frac{w_{1j}}{|w_j|}, \frac{w_{2j}}{|w_j|}, ..., \frac{w_{nj}}{|w_j|}],$$

$$|w_j| = \sqrt{w_{1j}^2 + w_{2j}^2 + ... + w_{nj}^2}\ ,$$

где i=2..12.

Рассмотрим отображение входного пространства образов для нормального состояния и атаки (тип атаки neptune) на плоскость двух первых главных компонент. Из рис. 6 видно, что данные, соответствующие одному классу атаки, могут концентрироваться в нескольких областях. Это затрудняет классификацию атак при использовании линейной RNN в силу сложных взаимосвязей, существующих между отдельными параметрами. Для устранения этого недостатка использовалась нелинейная RNN.

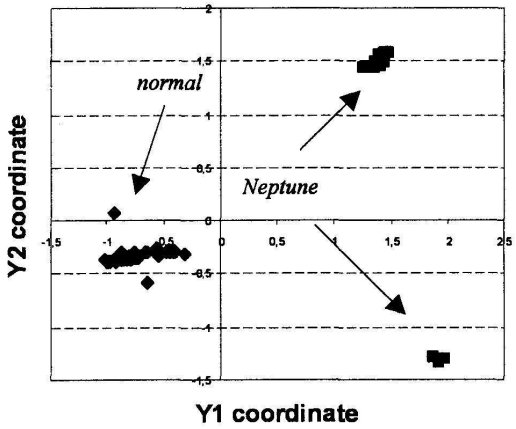

Рис. 6. *Данные, обработанные линейной RNN*

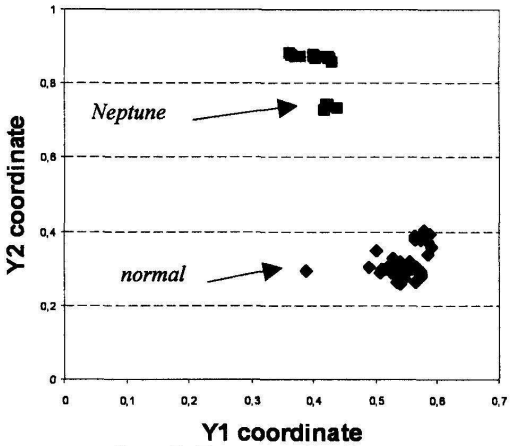

Рис. 7. *Данные, обработанные нелинейной RNN*

5. Ансамблевая нейронная сеть и MLP

Рассмотрим Ансамблевую нейронную сеть. Каждый эксперт представлен отдельной системой классификации (в нашем случае в качестве эксперта применена модель 1). Арбитр (многослойный персептрон), осуществляет процедуру голосования для формирования совместного решения всех трех экспертов.

Ансамблевая нейронная сеть обучается по алгоритму усиления за счет фильтрации [16], как показано на рис. 8. Этот алгоритм обучения состоит из следующих шагов:

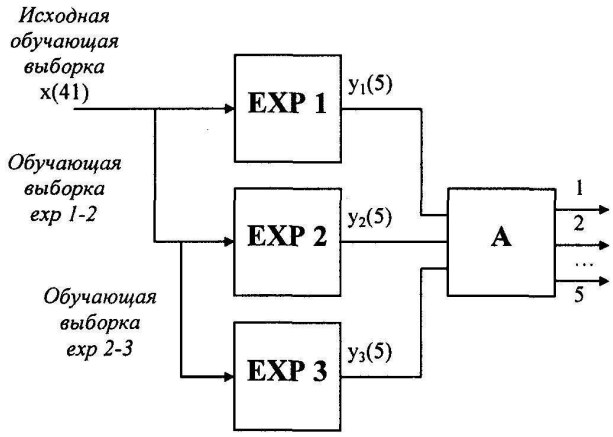

Рис. 8. *Третий вариант IDS (режим обучения)*

1) Первый эксперт обучается на исходном множестве примеров.

2) Обученный первый эксперт используется для фильтрации (filter) второго множества примеров. На этом этапе применяется процедура “подбрасывания монетки”.

(А) Исходное множество примеров последовательно “пропускается” через первого эксперта. Решение о включении очередного примера во второе множество делается на основании моделирования подбрасывания монетки. Если выпала “решка”, то корректно классифицированные примеры отклоняются до тех пор, пока не возникнет ошибка классификации. Пример, приведший к ошибке классификации, добавляется во множество примеров для обучения второго эксперта. Если выпал “орел”, то примеры отклоняются до тех пор, пока очередной пример не будет классифицирован правильно. Процедура “подбрасывания монетки” повторяется, пока количество примеров во второй выборке не станет равным их числу в первой. Если вследствие перебора достигнут последний элемент исходного множества примеров, то перебор повторяется с начала этого множества.

(Б) Отфильтрованное таким образом множество примеров подается для обучения второго эксперта.

Процедура “подбрасывания монетки” гарантирует, что при тестировании первого эксперта на втором наборе примеров ошибка классификации составит ½.

3) Множество примеров для обучения третьего эксперта формируется следующим образом.

(А) Пример из исходного множества “пропускается” через первого и второго экспертов. Если решения обоих экспертов совпадают, пример отклоняется; если они расходятся в своих мнениях, данный пример включается во множество примеров обучения третьего эксперта. Этот процесс продолжается до тех пор, пока не будет отфильтровано все исходное множество примеров.

(Б) Полученное множество используется для обучения третьего эксперта.

Как уже отмечалось, многослойный персептрон предназначен для классификации атак на основе главных компонент, полученных с выходов сети RNN (рис. 9). Количество нейро-

нов выходного слоя варьируется в зависимости от количества классов атак. Для обучения используется алгоритм обратного распространения ошибки.

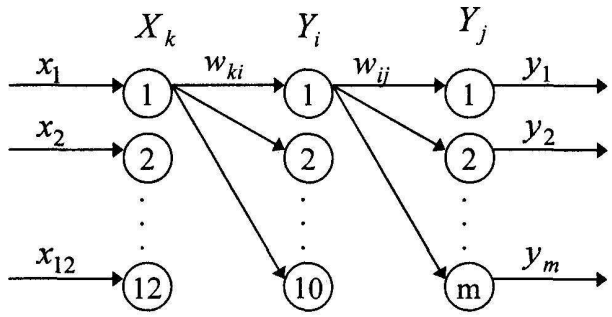

Рис. 9. *Архитектура MLP*

После обучения нейронных сетей они объединяются в единую систему обнаружения атак.

6. Результаты экспериментов

Чтобы оценить эффективность предложенных подходов обнаружения вторжений, был проведен ряд экспериментов. База данных KDD Cup 99 использовалась для обучения и тестирования нейросетевых моделей. Это одна из тех немногих баз в области обнаружения вторжений, которая привлекает внимание исследователей благодаря своей хорошо продуманной структуре и доступности.

Алгоритм усиления за счет фильтрации, который используется в случае модели 3, предполагает наличие большого (в идеале – бесконечного) множества примеров. Поэтому мы использовали 10% выборку из базы KDD (почти 500 000 записей!). Для обучения нейронных сетей были отобраны 6186 примеров. Далее вся 10% выборка применялась для тестирования. Те же наборы данных использовались для обучения и тестирования модели 1 и модели 2, что позволяет сравнивать производительности рассматриваемых в статье систем обнаружения атак друг с другом. Предлагаемые системы обнаружения вторжений осуществляют распознавание 5 классов атак, встречающихся в базе KDD, а именно: DoS, U2R, R2L, Probe и Normal.

Для изучения характеристик предложенных систем мы задались тремя основными показателями: доля обнаруженных, доля распознанных атак по каждому классу и число ложных срабатываний системы. Доля обнаруженных атак определяется как число образов атак отдельного класса, обнаруженных системой, деленное на общее количество записей об атаках этого класса в базе данных. Подобным образом определяется и доля распознанных. Ложные срабатывания указывают общее число образов нормальной работы сети, ошибочно классифицированных как атаки.

Рассмотрим функционирование системы на примере модели 1 (см. раздел 3). Эта модель достаточно проста. Результаты тестирования в режиме распознавания класса атаки приведены в таблице 1.

Таким образом, наилучший результат был достигнут для атак класса DoS и Probe (почти однозначная распознаваемость). Несколько хуже определяются U2R и R2L, соответственно 80,77% и 58,44%. Кроме того, существует процент ложных срабатываний системы.

Недостатком модели 1 является большое число ложных срабатываний. С целью улучшения этого показателя были использованы более сложные модели, описанные в разделе 3. Модель 2, благодаря применению отдельного персептрона для каждого класса атак, допускает меньше ошибок при обработке образов, соответствующих “нормальному” режиму функционирования сети. В модели 3 окончательное решение формируется с учетом мнения трех экспертов. Как уже упоминалось выше, каждый эксперт представлен отдельной системой классификации (в экспериментах в качестве эксперта применена модель 1). Т.е. каждый последующий эксперт корректирует заключение предыдущих, формируя тем самым общее решение нескольких нейронных сетей.

Таблица 1. *Результаты тестирования модели 1*

класс	всего	обнаружено	распознано
DoS	391458	391441 (99.99%)	370741 (94.71%)
U2R	52	48 (92.31%)	42 (80.77%)
R2L	1126	1113 (98.85%)	658 (58.44%)
Probe	4107	4094 (99.68%)	4081 (99.37%)
normal	97277	---	50831 (52.25%)

Таблица 2. *Результаты тестирования модели 2*

класс	всего	обнаружено	распознано
DoS	391458	391063 (99.90%)	370544 (94.66%)
U2R	52	49 (94.23%)	37 (71.15%)
R2L	1126	1088 (96.63%)	1075 (95.47%)
Probe	4107	3749 (91.28%)	3735 (90.94%)
normal	97277	---	83879 (86.22%)

Таблица 3. *Результаты тестирования модели 3*

класс	всего	обнаружено	распознано
DoS	391458	391443 (99.99%)	370663 (94.69%)
U2R	52	50 (96.15%)	42 (80.76%)
R2L	1126	1102 (97.87%)	1086 (96.45%)
Probe	4107	3954 (96.27%)	3939 (95.91%)
normal	97277	---	84728 (87.09%)

Результаты тестирования (таблица 2 и таблица 3) мало отличаются, поэтому тяжело было сравнивать эти две модели между собой. Но при ближайшем рассмотрении выбор был сделан в пользу модели 3.

Сводные данные по каждому из вариантов построения системы обнаружения атак приведены в таблице 4.

Таким образом, модель 3 характеризуется высокой точностью (93,21%) и наименьшим числом ложных срабатываний. При использовании модели 1 были распознаны 86,3% входных образов, а модели 2 – 92,97%. Модели 2 и 3 могут успешно применяться для работы с большими наборами сложных по структуре данных.

Рассмотрим случай с нелинейной RNN в модели 1. Нелинейная RNN использовалась с сигмоидной функцией активации. Мы применяли модель 1 с линейным и нелинейным модулем RNN к отдельным службам (HTTP, FTP_DATA, SMTP и т.д.) (см. таблица 5 и таблица 6).

Очевидно, что однозначного ответа на вопрос – что предпочтительнее – нет. Для службы HTTP желательно применять нелинейную RNN, в отличие от FTP_DATA, где линейная RNN демонстрирует лучшие результаты. Поэтому выбор в пользу одного или другого варианта необходимо делать для каждого случая отдельно.

Таблица 4. *Сводные данные по результатам тестирования каждой модели*

модель	Обнаруженные атаки	Распознанные атаки	Ложные срабатывания	Общая доля распознанных %
модель 1	396696 (99.98%)	375522 (94,65%)	46446 (47.75%)	86.30%
модель 2	395949 (99.80%)	375391 (94.61%)	13398 (13.77%)	92.97%
модель 3	396549 (99.95%)	375730 (94.70%)	12549 (12.90%)	93.21%

Таблица 5. *Обнаружение и распознавание атак для службы HTTP* (*модель 1*)

модель	служба HTTP			
	Обнаруженные атаки	Распознанные атаки	Ложные срабатывания	Общая доля распознанных %
линейная RNN	2407 (100%)	2406 (99.96%)	470 (0.76%)	99.27%
нелинейная RNN	2407 (100%)	2405 (99.92%)	65 (0.11%)	99.92%

Таблица 6. *Обнаружение и распознавание атак для службы FTP_DATA* (*модель 1*)

модель	служба FTP_DATA			
	Обнаруженные атаки	Распознанные атаки	Ложные срабатывания	Общая доля распознанных %
линейная RNN	893 (96.74%)	881 (95.45%)	76 (2.00%)	97.50%
нелинейная RNN	866 (93.82%)	400 (43.34%)	44 (1.16%)	87.99%

7. Заключение

В работе рассмотрены различные подходы к построению систем обнаружения атак, которые базируются на нейросетевых технологиях. Путем комбинирования двух нейронных сетей, а именно RNN и MLP, можно идентифицировать и распознавать атаки на компьютерные сети с достаточно высокой степенью точности. В качестве базы данных для тестирования предложенных методов использовалась база KDD-99. Основными преимуществами применения подходов, основанных на нейронных сетях, являются способность адаптироваться к динамическим условиям и быстрота функционирования, что особенно важно при работе системы в режиме реального времени.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. S.Kumar and E.H.Spafford. A Software architecture to support misuse intrusion detection // Proceedings of the 18th National Information Security Conference. - 1995. - P. 194-204.
2. K.Ilgun, R.A.Kemmerer, P.A.Porras. State transition analysis: A rule-based intrusion detection approach // IEEE Transaction on Software Engineering. - 1995. - Vol. 21, N 3. - P. 181-199.
3. SNORT, http://www.snort.org.
4. T.Lunt, A.Tamaru, F.Gilham, et al. A Real-time Intrusion Detection Expert System (IDES) – final technical report // Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, Feb. 1992.
5. P.A.Porras and P.G.Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances // Proceedings of National Information Systems Security Conference. - Baltimore MD, 1997.
6. D.E.Denning. An intrusion-detection model // IEEE Transaction on Software Engineering. - 1987. - Vol. 13, N 2. - P. 222-232.
7. W.Lee, S.Stolfo, K.Mok. A data mining framework for adaptive intrusion detection // Proceedings of the 1999 IEEE Symposium on Security and Privacy. - Los Alamos, CA, 1999. - P. 120-132.
8. W.Lee, S.Stolfo. A Framework for constructing features and models for intrusion detection systems // ACM Transactions on Information and System Security. - 2000. - Vol. 3, N 4. - P. 227-261.
9. Y.Liu, K.Chen, X.Liao, et al. A genetic clustering method for intrusion detection // Pattern Recognition. - 2004. - Vol. 37, N 5. - P. 927-924.
10. E.Eskin, A.Rnold, M.Prerau, L.Portnoy, S.Stolfo. A Geometric framework for unsupervised anomaly detection // Applications of Data Mining in Computer Security, Kluwer Academics. - 2002.
11. M.Shyu, S.Chen, K. Sarinnapakorn, L.Chang. A Novel Anomaly Detection Scheme Based on Principal Component Classifier // Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03). - 2003. - P. 172-179.
12. H.Kayacik, A.Zincir-Heywood and M.Heywood. On the capability of an SOM based intrusion detection system // in Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN'03). - 2003. - P. 1808-1813.
13. Zheng Zhang, Jun Li, C.N. Manikopoulos, Jay Jorgenson, Jose Ucles. HIDE : a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification // Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy. - West Point, NY, 2001. - P. 85-90.
14. V.Golovko, L.Vaitsekhovich. Neural Network Techniques for Intrusion Detection // Proceedings of International Conference on Neural Networks and Artificial Intelligence (ICNNAI-2006). - 2006. - P. 65-69.
15. 1999 KDD Cup Competition, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
16. H.Drucker, R.Schapire and P.Simard. Improving performance in neural networks using a boosting algorithm // In S.J.Hanson, J.D.Cowan and C.L.Giles eds., Advanced in Neural Information Processing Systems 5, Denver, CO, Morgan Kaufmann, San Mateo, CA. - 1993. - P. 42-49.
17. E. Oja. Principal components, minor components and linear networks // Neural Networks. - 1992. - Vol. 5. - P. 927-935.

Статья поступила в редакцию 07.12.2006