УДК 004.056

Методы интеллектуального анализа данных и обнаружение вторжений

Е. В. Зубков, В. М. Белов

В работе рассматривается проблематика использования методов интеллектуального анализа данных (Data Mining) (МИАД) для решения задач обнаружения вторжений. Этот вопрос является ключевым при построении систем обнаружения вторжений (Intrusion Detection Systems – IDS), в том числе и сетевых (Network IDS – NIDS), в основе которых лежит принцип выявления аномалий. В статье дается обзор наиболее популярных подходов, используемых для обнаружения вторжений, и приводятся примеры приложений.

Ключевые слова: интеллектуальный анализ данных (ИАД), информационная безопасность (ИБ), обнаружение вторжений, аномалии, наивный байесовский подход (НБП), метод опорных векторов (МОВ), метод ближайших соседей (МБС), метод деревьев решений (МДР), искусственные нейронные сети (ИНС), нечеткая логика (НЛ), генетические алгоритмы (ГА).

1. Введение

Среди ключевых задач в парадигме обеспечения ИБ выделяют противодействие компьютерным атакам. Эта цель достигается за счет комплексного применения ряда технических средств, к числу которых относятся и IDS.

Значительные усилия исследователей сосредоточены на разработке новых подходов при построении сетевых IDS – NIDS, когда источником исходных данных является сетевой трафик. Модель сетевой активности, построенная на основе атрибутов сетевых пакетов (как правило, сетевого и транспортного уровня), обрабатывается определенным образом. В результате формируется профиль нормального сетевого состояния. В дальнейшем всякое отличие от полученного профиля (аномалия) рассматривается как потенциально опасное событие.

Для сравнения: в сигнатурных методах, наоборот, создаются профили компьютерных атак (сигнатуры). Вредоносными считаются события, соответствующие сигнатурам. Прочий трафик считается легитимным.

Подход на основе выявления аномалий алгоритмически является более сложной и трудоемкой задачей, но в то же время и более перспективной, поскольку позволяет выявлять ранее неизвестные атаки. Рост вычислительных мощностей и снижение их удельной стоимости на современном этапе развития является фактором, сопутствующим развитию IDS именно в этом направлении. В качестве алгоритмической основы используются так называемые МИАД. Их суть в общем случае заключается в обработке больших объемов данных с целью обнаружения полезной, но не очевидной информации об этих данных.

В статье также рассматриваются наиболее популярные МИАД, применяемые для обнаружения вторжений, и примеры исследований, проводимых в данной предметной области.

2. Основные МИАД

2.1. Наивный байесовский подход (Naive Bayes Approach)

Так называемый НБП является наиболее простым вариантом метода, использующего байесовские сети. Его применяют в задачах классификации [1]. Термин «наивный» связан с предположением, что все рассматриваемые переменные независимы друг от друга, что сильно упрощает вычисления по сравнению с тем, если бы значения переменных были математически зависимы [2, 3]. И хотя это предположение не всегда выполняется, на практике данный алгоритм находит широкое применение. Обозначим $P(c_j)$ как вероятность того, что некоторый объект a_i относится к классу c_j. Пусть объект a_i характеризуется набором независимых переменных (v_1, v_2 … v_m). Событие, соответствующее равенству независимых переменных определенным значениям ($v_1 = v_1^a$ и $v_2 = v_2^b$ и … и $v_m = v_m^c$), обозначим как E, соответственно, вероятность его наступления будет равна $P(E)$.

Идея алгоритма заключается в расчете условной вероятности принадлежности объекта a_i к c_j при наступлении события E.

Из теории вероятностей известно, что ее можно вычислить по формуле:

$$P(c_j \mid E) = \frac{P(E \mid c_j) \cdot P(c_j)}{P(E)}.$$

Априорные вероятности $P(c_j)$ – вероятности того, что произвольно взятый элемент a_i относится к классу c_j – вычисляются на основании информации обучающего множества как отношение числа элементов, принадлежащих данному классу, к общему количеству элементов в нем:

$$P(c_j) = \frac{N}{n_j}.$$

Следующим шагом является вычисление для каждого признака v_i условных вероятностей появления каждого его возможного значения относительно заданных классов. Иными словами, имея множество возможных значений каждого признака $v_i = \{v_i^1, v_i^2, \ldots, v_i^q\}$, вычисляют вероятности появления элемента с определенным значением признака $v_i = v_i^r$ в каждом классе c_j:

$$P(v_i^r \mid c_j) = \frac{n^{i,r}}{n_j^{i,r}},$$

где $n^{i,r}$ и $n_j^{i,r}$ – общее количество элементов, у которых признак $v_i = v_i^r$, и количество таких элементов, попавших в класс c_j, соответственно.

В результате для каждого класса получим множество (по количеству возможных событий E) записей вида: если $v_1 = v_1^a$ и $v_2 = v_2^b$ и … и $v_m = v_m^c$, то элемент принадлежит классу c_j с вероятностью $P(c_j|E)$.

Максимальное значение вероятности $P_{\max} = \max P(c_j|E)$ определяет правило, по которому элемент будет относиться к своему классу.

В работах [4] и [1] со ссылкой на [5–7] отмечают следующие отличительные качества НБП.

Преимущества:

- простота в использовании;
- высокая скорость – классификация данных осуществляется за одно сканирование;

– способность обрабатывать отсутствующие значения атрибутов (при расчете достоверности каждого класса вероятности отсутствующих значений просто не учитываются);

– поскольку в модели определяются зависимости между всеми переменными, легко обрабатываются ситуации, когда значения некоторых переменных неизвестны;

– построенные байесовские сети просто интерпретируются и позволяют на этапе прогностического моделирования легко производить анализ по сценарию "что - если";

– подход позволяет естественным образом совмещать закономерности, выведенные из данных, и фоновые знания, полученные в явном виде (например, от экспертов);

– использование байесовских сетей позволяет избежать проблемы переобучения (over fitting), то есть избыточного усложнения модели, чем страдают многие методы при слишком буквальном следовании распределению зашумленных данных.

Преимущества прикладного применения метода в задачах обнаружения вторжений приведены в [8].

Недостатки:

– перемножение условных вероятностей корректно только при действительной статической независимости входных переменных; несмотря на неплохие практические результаты допущения этой независимости (чем обусловлена приставка «наивный» в названии), необходимо учитывать, что корректно данная ситуация обрабатывается более сложными методами.

– невозможна непосредственная обработка непрерывных переменных – их требуется разбивать на множество интервалов, чтобы атрибуты были дискретными; такое разбиение в ряде случаев приводит к потере значимых закономерностей;

– НБП учитывает только индивидуальное влияние входных переменных на результат классификации, не принимая во внимание комбинированного влияния пар или троек значений разных атрибутов, что было бы полезно с точки зрения прогностической точности, но значительно увеличило бы количество проверяемых комбинаций.

2.2. Метод опорных векторов

Исходными данными в МОВ является множество элементов, размещаемых в пространстве. Размерность пространства соответствует количеству классифицирующих признаков, а их значение определяет положение элементов (точек) в пространстве. Основная идея МОВ заключается в переводе исходных векторов в пространство более высокой размерности и построении разделяющей гиперплоскости [9–11]. Если обучающие данные являются линейно разделяемыми, то можно построить дополнительно две гиперплоскости, параллельные разделяющей, таким образом, чтобы точки обоих классов, ближайшие к разделяющей гиперплоскости (опорные векторы), принадлежали этим гиперплоскостям. Они образуют полосу, свободную от объектов классификации (точек). Дальнейшая задача состоит в том, чтобы достичь максимального значения ширины этой полосы [12]. Разделяющую гиперплоскость в этом случае называют оптимальной (рис. 2.1).

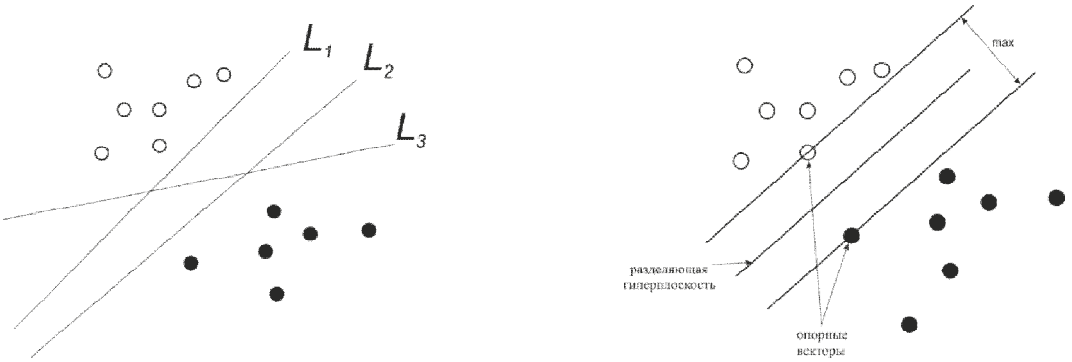

Рис. 2.1. Примеры классификации элементов на плоскости: произвольная (слева), оптимальная (справа)

В случае если множество не является линейно разделимым, применяют увеличение размерности пространства. Затем в уже измененном пространстве находят линейный разделитель. Построение разделяющей гиперплоскости в этом случае требует применения специальной нелинейной функции, которую называют ядром [13].

Отмечают следующие преимущества МОВ:

– МОВ считают наиболее быстрым методом нахождения решающих функций;

– метод сводится к решению задачи квадратичного программирования в выпуклой области, которая всегда имеет единственное решение;

– метод находит разделяющую полосу максимальной ширины, что позволяет в дальнейшем осуществлять более уверенную классификацию;

Из недостатков отмечают:

– метод подходит только для решения задач с двумя классами;

– чувствительность к шумам и стандартизации данных;

– отсутствие общего подхода к автоматическому выбору ядра (и построению спрямляющего подпространства в целом) в случае линейной неразделимости классов.

2.3. Метод ближайших соседей (Nearest Neighbor)

МБС является самым простым алгоритмом классификации [14, 15]. Он относит классифицируемый объект $a \in A$ к тому классу $c \in C$, которому принадлежит ближайший обучающий объект $a' \in A'$.

Сформулируем основные понятия данного подхода. Существует пространство объектов $A = \{a_1, a_2, \dots, a_n\}$ и конечное множество классов $C = \{c_1, c_2, \dots, c_k\}$. Кроме того, определена обучающая выборка $A' \subset A$, для которой известна целевая зависимость $A' \to C$: $c_i = c'(a'_i)$. Задача алгоритма – аппроксимировать целевую зависимость $c'(a'_i)$ на всем множестве A.

Упорядочим элементы обучающего множества A' относительно произвольного объекта $a \in A$ в порядке возрастания расстояния от $a'_{i,a}$ до a:

$$a'_{1,a}, a'_{2,a}, \dots, a'_{m,a},$$

тогда согласно алгоритму объект a будет отнесен к тому же классу, что и ближайший к нему объект обучающей выборки $a'_{1,a}$:

$$c(a; A') = c_{1,a}.$$

Как справедливо отмечают в [14], единственным преимуществом этого метода является его простота. Естественным развитием описанного алгоритма стал алгоритм k-ближайших соседей. Чтобы сгладить шумовое влияние выбросов, классификацию объектов осуществляют по k-ближайшим соседям. Каждый из соседей $a'_{i,a}, i = 1,2, \dots, k$ голосует за отнесение объекта a к своему классу $c_{i,a}$. Алгоритм относит объект a к тому классу, который наберет большее число голосов. Голосование может быть невзвешенным и взвешенным [16].

Преимущества:

– процесс обучения заключается в запоминании обучающей выборки;

– простота реализации и возможность вводить дополнительные параметры настройки;

– прецедентная логика работы алгоритма хорошо понятна экспертам в предметных областях (медицина, биометрия, юриспруденция);

Недостатки:

– приходится хранить обучающую выборку целиком, отсюда и неэффективный расход памяти.

– большое количество операций при классификации объектов [17].

2.4. Метод построения деревьев решений (Decision Trees)

МДР является одним из наиболее популярных методов решения задач классификации и прогнозирования. Иногда этот метод из МИАД также называют деревьями решающих правил, деревьями классификации и регрессии [18-21].

Структурно дерево состоит из элементов трех категорий:

– узлы, не являющиеся листьями, – это атрибуты, по которым различают элементы, подлежащие классификации;

– листья, являющиеся метками, – со значениями решений для классификации данных;

– ребра – значения атрибута, из которого исходит ребро.

Процесс классификации осуществляется путем последовательного следования по дереву сверху вниз. На каждом уровне дерева решение принимается на основе значений атрибутов. Обычно каждый узел включает проверку одной независимой переменной. Иногда в узле дерева две независимые переменные сравнивают друг с другом или определяют некоторую функцию от одной или нескольких переменных. Если значением переменной является число, то проверяют больше или меньше это значение некоторой константы. Иногда область числовых значений разбивают на интервалы и проверяют попадание значения в один из них. Результат оценки всегда соответствует только одному из ребер, исходящих из узла принятых решений [22].

Для создания дерева используют алгоритмы итеративного построения. Обычно создание оптимального дерева вычислительно невозможно, поскольку количество возможных деревьев растет экспоненциально в зависимости от набора признаков. Существует несколько подходов к решению данной проблемы, основанных на минимизации энтропии [8, 23]. Фактически задача сводится к последовательному определению очередного атрибута для деления множества элементов. При этом могут использовать различные критерии: прирост информации (Information Gain), коэффициент усиления (Gain Ratio) [24], индекс Джини (Gini), Хи-квадрат, G-квадрат [25].

Среди основных преимуществ МДР отмечают [8, 21, 26]:

– Достаточно наглядное представление решения даже для непрофессионального пользователя.

– Возможность преобразования результата в набор булевых правил, которые просты для встраивания в технологии реального времени, например, IDS и МЭ.

– Возможность работы как с числовыми, так и с номинальными атрибутами.

– Относительно высокую скорость работы.

– Использование модели «белого ящика». Если определенная ситуация наблюдается в модели, то её можно объяснить при помощи булевой логики.

– Возможность обработки данных, которые содержат ошибочные или пропущенные значения.

Недостатки:

– Большинство алгоритмов (например, ID3 и C4.5) требуют, чтобы целевые атрибуты принимали только дискретные значения.

– Поскольку основным принципом деревьев решений является «разделяй и властвуй», то они, как правило, хорошо работают, когда исследуемое множество содержит несколько значимых признаков, и значительно хуже, когда между элементами существуют сложные взаимосвязи (получаются слишком сложные конструкции, которые недостаточно полно описывают данные).

– Высокая зависимость результата от качества обучающей выборки. Наличие шума может привести к выбору неоптимального признака при делении. Если такое деление происходит близко к корню, это приводит к усложнению общей структуры дерева и многократному дублированию отдельных сегментов.

– Близорукий характер работы большинства индукционных алгоритмов. Неспособность смотреть более чем на один шаг вперед, а значит, неспособность строить критерии на основе комбинации признаков. Использование стратегий с более глубоким тестированием приводит к значительному увеличению вычислительной нагрузки и не дает ожидаемого полезного эффекта.

– Неэффективны при решении задач классификации с большим числом классов.

Результаты сравнительного анализа алгоритмов на основе деревьев решений по отношению к другим алгоритмам приведены в работе [27].

2.5. Искусственные нейронные сети

ИНС можно отнести к наиболее интенсивно развивающемуся направлению предметной области, связанной с МИАД. Большое количество работ по данной тематике связано, прежде всего, с универсальностью, которая изначально была заложена в ИНС [28–32]. В настоящее время ИНС применяются для решения широкого спектра задач (классификация образов, кластеризация, аппроксимация функций, прогноз, оптимизация, организация памяти, адресуемой по содержанию, управление).

ИНС являются предельно упрощенными аналогами естественных нейронных сетей. Элементарной структурной единицей ИНС является искусственный нейрон [28, 29, 33]. В состав нейрона входят умножители (синапсы), сумматор и нелинейный преобразователь (рис. 2.2).

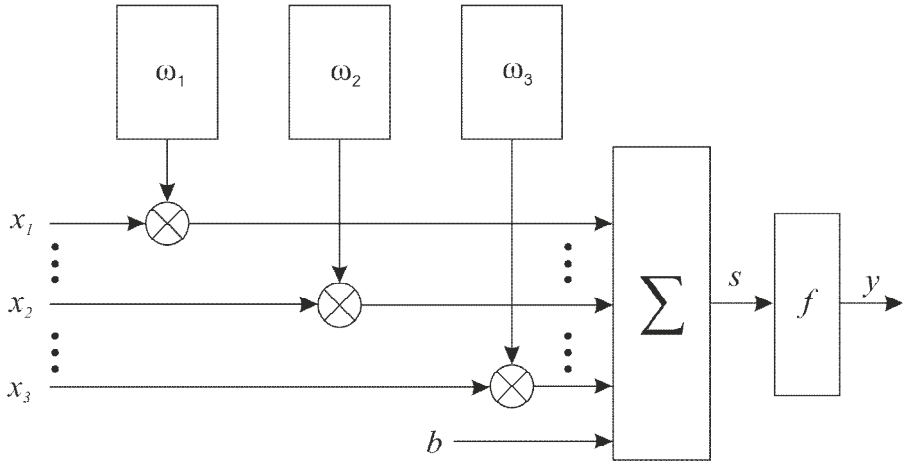

Рис. 2.2. Структура искусственного нейрона

Синапсы осуществляют связь между нейронами и умножают входной сигнал на вес синапса . Сумматор выполняет сложение входных сигналов и смещения на величину , Нелинейный преобразователь реализует нелинейную функцию одного аргумента – выхода сумматора . Эту функцию называют «функцией активации» или «передаточной функцией» нейрона. Она может быть: пороговой, знаковой, сигмоидальной, полулинейной, линейной и т.д. [28, 31, 32, 34]. Математически нейрон можно описать следующим образом:

$$S = \sum_{i=1}^{n} \omega_i x_i + b\,.$$

В настоящее время существует несколько десятков различных нейросетевых архитектур. Однако практически все они связаны с выбором и анализом некоторых частных видов структур с известными свойствами (сети Хопфилда, Гроссберга, Кохонена) [28, 34]. Наиболее популярными и изученными являются следующие: многослойный персептрон, сети Кохонена, нейронные сети встречного распространения, сети Хопфилда и Хэмминга, сеть с радиальными базисными элементами (RBF), вероятностная нейронная сеть (PNN), обобщенно-регрессионная нейронная сеть (GRNN), линейные нейронные сети.

Использование существующих ИНС открывает перед исследователями широкие возможности в прикладных областях, но вместе с тем оно связано и с рядом проблемных вопросов [1, 28, 30, 35].

Преимущества ИНС:

– Приобретение в процессе обучения способности отображения входной информации в выходную информацию без использования каких-либо сведений о статистической модели данных (вероятностной модели распределения).

– Возможность расширения функциональных возможностей ИНС за счет применения нелинейных искусственных нейронов.

– Способность к адаптации и изменению внешних условий путем переобучения. Возможность построения ИНС, изменяющих свои характеристики с течением времени и способных работать в нестационарной (nonstationary) среде, где статистика изменяется с течением времени.

– Параллельная структура нейронных сетей потенциально ускоряет решение некоторых задач и обеспечивает масштабируемость ИНС.

– Универсальность механизма ИНС обеспечивает возможность применения одного проектного решения в нескольких предметных областях.

– Способность элементарно учитывать наблюдаемые или интуитивно предполагаемые поправки, требующие огромных вычислений и расчетов.

Говоря о недостатках ИНС, отмечают следующее:

– Отсутствие строгой теории по выбору структуры ИНС.

– В силу своей природной универсальности ИНС могут обучиться достаточно сложным закономерностям. Обратная сторона этого свойства заключается в том, что количество степеней свободы может превышать число использовавшихся для обучения примеров. Это значит, что в принципе ИНС можно обучить даже на случайно сгенерированном массиве чисел. В работе [1] со ссылкой на [36] приводится пример, когда использование ИНС позволяет объяснить историю прошлых событий, но не дает обоснованного прогноза на будущее.

– Обученные нейронные сети являются нетрактуемыми моделями – «черными ящиками», поэтому логическая интерпретация описанных ими закономерностей практически невозможна (за исключением простейших случаев).

– Возможность обработки только численных переменных. Соответственно при работе с переменными других типов возникает необходимость их числового кодирования. При этом, как отмечается в [1] со ссылкой на [37], необходимо ввести по новой переменной для каждого значения исходной. Таким образом, при большом количестве нечисловых переменных с большим количеством возможных значений использование нейронных сетей становится совершенно невозможным.

2.6. Нечеткая логика

Математическая теория нечетких множеств (Fuzzy Sets) (ТНМ) и НЛ (Fuzzy Logic) являются обобщениями классической теории множеств и классической формальной логики. Данные понятия были впервые предложены американским ученым Лотфи Заде (Lotfi Zadeh) в 1965 г. Основной причиной появления новой теории стало наличие нечетких и приближенных рассуждений при описании человеком процессов, систем, объектов [38, 39, 40].

Под нечетким множеством понимается множество пар вида

$$C = \{x / MF_C(x)\},$$

где $x \in X$ – некоторый элемент (параметр), а $MF_C(x)$ – функция принадлежности (Membership Function) (ФП). ФП является основной характеристикой НМ и определяет степень принадлежности элемента a множеству C. ФП может принимать значения на интервале от 0 до 1. Значение $MF_C(x) = 1$ означает полную принадлежность, $MF_C(x) = 0$ – отсутствие

таковой. В качестве популярного примера можно привести нечеткое множество «горячий чай», где параметром оценки будет температура, измеренная в градусах Цельсия. Тогда:

Известно достаточно большое количество типовых ФП. Наибольшее распространение получили: треугольная, трапецеидальная и гауссова ФП [38, 41, 42]. Их примеры изображены на рис. 2.3.

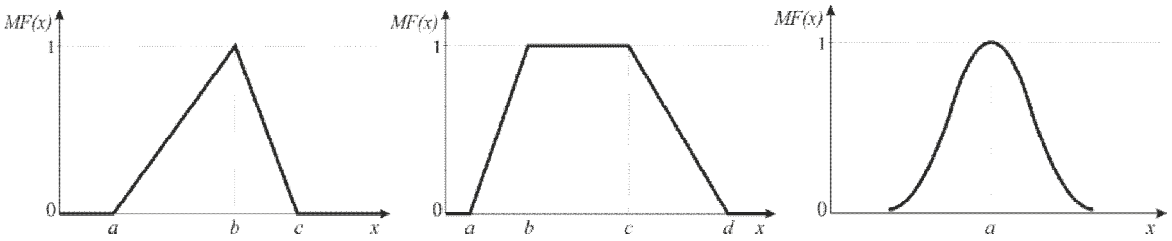

Рис. 2.3. Функции принадлежности: треугольная, трапецеидальная, гауссова соответственно

В настоящее время НЛ успешно применяют в комбинации с другими подходами МИАД, образуя эффективные гибридные МИАД. Среди наиболее интересных примеров можно отметить:

– Нечеткие ИНС – формируют выводы на основе аппарата НЛ. Параметры ФП настраивают с использованием алгоритмов обучения ИНС. Нечеткие ИНС характеризуются быстрыми алгоритмами обучения и интерпретируемостью накопленных знаний.

– Адаптивные нечеткие системы – позволяют исключить участие экспертов на этапе построения правил и ФП. Требуемый результат достигают за счет применения алгоритмов обучения на экспериментальных данных. Несмотря на относительную сложность и трудоемкость обучения по сравнению с алгоритмами обучения ИНС, эти решения активно применяют на практике, например, – в совокупности с ГА.

– Нечеткая кластеризация – в отличие от четких методов позволяет одному и тому же объекту принадлежать одновременно нескольким кластерам, но с различной степенью принадлежности. Нечеткая кластеризация во многих ситуациях более «естественна», чем четкая, например, для объектов, расположенных на границе кластеров.

Выделяют следующие преимущества НЛ [39]:

– формализуется и объединяется опыт операторов и разработчиков в настройке петель регулирования;

– предлагается простой метод управления для сложных процессов;

– постоянно учитывается опыт по управлению процессами данного типа, принимая во внимание исключения разного рода и особенности системы;

– учитываются исходные данные разного рода и производится объединение разных исходных данных;

– интегрируется с МИАД, помогает в создании более абстрактных и гибких шаблонов для обнаружения вторжений [48].

К недостаткам НЛ относят следующие моменты [28]:

– исходный набор нечетких правил формулируется экспертом-человеком и может оказаться неполным или противоречивым;

– вид и параметры функций принадлежности, описывающих входные и выходные переменные системы, выбираются субъективно и могут оказаться не вполне отражающими реальную действительность.

2.7. Генетические алгоритмы

ГА относят к числу универсальных методов оптимизации, позволяющих решать задачи различных типов (комбинаторные, общие задачи с ограничениями и без ограничений) и различной степени сложности [3, 43–45].

ГА являются стохастическим эвристическим методом, в котором вероятность выбора состояния $S(t+1)$ зависит от состояния $S(t)$ и косвенно от предыдущих состояний.

Так, в работе [43] используют следующие понятия при описании генетических алгоритмов.

Хромосома – вектор (или строка) из каких-либо чисел. Если этот вектор представлен бинарной строкой из нулей и единиц, например, 1010011, то он получен либо с использованием двоичного кодирования, либо кода Грея. Каждая позиция (бит) хромосомы называется геном.

Индивидуум (генетический код, особь) – набор хромосом (вариант решения задачи). Обычно особь состоит из одной хромосомы, поэтому в дальнейшем особь и хромосома – идентичные понятия.

Расстояние – хеммингово расстояние между бинарными хромосомами.

Кроссинговер (кроссовер) – операция, при которой две хромосомы обмениваются своими частями. Например, $1100\&1010 \rightarrow 1110\&1000$.

Мутация – случайное изменение одной или нескольких позиций в хромосоме. Например, $1010011 \rightarrow 1010001$.

Инверсия – изменение порядка следования битов в хромосоме или в ее фрагменте. Например, $1100 \rightarrow 0011$.

Популяция – совокупность индивидуумов.

Пригодность (приспособленность) – критерий или функция, экстремум которой следует найти.

Основные принципы работы ГА заключены в следующем алгоритме:

1. Генерация начальной популяции из n хромосом.
2. Вычисление для каждой хромосомы ее пригодности.
3. Выбор пары хромосом-родителей с помощью одного из способов отбора.
4. Применение операции кроссинговера для двух родителей с вероятностью p_c для получения двух потомков.
5. Проведение мутации потомков с вероятностью p_m.
6. Повторение шагов 3–5, пока не будет сгенерировано новое поколение популяции, содержащее n хромосом.
7. Повторение шагов 2–6, пока не будет достигнут критерий окончания процесса.

Критерием окончания процесса может служить заданное количество поколений или схождение (*convergence*) популяции. Схождением называют такое состояние популяции, когда все строки популяции почти одинаковы и находятся в области некоторого экстремума. В такой ситуации кроссинговер практически никак не изменяет популяции, так как создаваемые при нем потомки представляют собой копии родителей с перемененными участками хромосом. Вышедшие из этой области за счет мутации особи склонны вымирать, так как чаще имеют меньшую приспособленность, особенно если данный экстремум является глобальным максимумом. Таким образом, схождение популяции обычно означает, что найдено лучшее или близкое к нему решение.

Несмотря на уникальные способности ГА для решения задач глобальной оптимизации в сравнении с другими техниками, исследователи отмечают тот факт, что ГА требуют значительных усилий при настройке под конкретную задачу. Прежде всего, в настройке нуждаются вероятности применения генетических операторов P_c, P_m [44].

Кроме того, выделяют ряд характерных свойств ГА [43].

Достоинства:

– Не требуют никакой информации о поведении функции (например, дифференцируемости и непрерывности).
– Относительно стойки к попаданию в локальные оптимумы.
– Пригодны для решения крупномасштабных проблем оптимизации.
– Могут быть использованы для широкого класса задач.
– Просты в реализации.

Недостатки:

– С помощью ГА проблематично найти точный глобальный оптимум.
– ГА непросто смоделировать для нахождения всех решений задачи.
– Не для всех задач удается найти оптимальное кодирование параметров.
– Дополнительный шум существенно замедляет поиск решения.

3. Исследования в области интеллектуального анализа данных для выявления вторжений

Направление исследований, связанных с применением data mining методик для выявления вторжений, представляется весьма перспективным. Снижение уровня ложных срабатываний отмечается практически всеми авторами в числе приоритетных задач. О масштабах исследований можно судить даже по работам, размещенным в открытом доступе. Ниже представлен краткий обзор некоторых из них.

Для отладки своих решений авторы использовали KDD Cup 99 Data (KDD99) [46]. KDD99 – это набор данных, который был сгенерирован путем эмуляции военного сетевого окружения в 1999 г. Он содержит в общей сложности 24 типа атак в обучающем наборе данных и еще 14 дополнительных типов в тестовых данных – в общей сложности чуть меньше 5 млн. записей.

Применение какого-либо метода в чистом виде встречается достаточно редко. Как правило, авторы применяют их в комбинации с целью получения определенного положительного эффекта. Так, в [4] авторы предлагают использовать традиционный байесовский подход в сочетании с бустингом.

Под термином «бустинг» понимается итеративный процесс, адаптивно изменяющий распределение обучающих примеров таким образом, что базовые классификаторы будут сосредоточены на тех примерах, которые трудно классифицировать. В предложенном решении в качестве базового классификатора используют НБП. На каждом этапе элементам обучающей выборки присваивают весовой коэффициент. У правильно классифицированных элементов вес будет уменьшен, для ошибочно классифицированных – увеличен. Таким образом, вес элемента будет тем меньше, чем меньше значение ошибки классификации. И наоборот, вес будет больше у трудно классифицируемых данных, имеющих высокое значение ошибки классификации. Процесс обучения заканчивается, когда все элементы классифицированы правильно или после определенного количества циклов. Авторы предоставляют результаты сравнительного анализа в пользу своего метода по сравнению с другими подходами (k-Nearest-Neighbor classifier (kNN), Decision Tree Classifier (C4.5), Support Vector Machines (SVM), Neural Network (NN), and Genetic Algorithm (GA)) и дают ему высокую оценку. Согласно приведенным данным предложенный метод лидирует по всем оценочным параметрам. Однако, как отмечают сами авторы, основной проблемой бустинга является переобучение. Таким образом, можно предположить, что высокие показатели могут быть следствием переобучения системы. Для более объективной оценки эффективности данного метода полезно увидеть, как алгоритм работает с ранее неизвестными данными.

Эти же авторы в работе [47] предлагают обучающийся алгоритм для адаптивного выявления сетевых вторжений, построенный на основе НБП и МДР. Рассматривают проблемы

работы с зашумленными данными, непрерывными атрибутами выбора набора входных атрибутов.

Другой пример использования МИАД – это совмещение НЛ, ассоциативных правил и ГА [48]. Предпосылками к использованию НЛ авторы считают два момента. Во-первых, при обнаружении компьютерного вторжения необходимо учитывать много статистических характеристик (время использования CPU, продолжительность соединения, длину пакета, количество различных TPC/UDP сервисов и т.д.), которые потенциально могут быть рассмотрены как нечеткие. Вторая мотивация заключается в решении проблемы, когда безопасность сама по себе содержит нечеткость. Ассоциативные правила, используемые в данном подходе, позволяют описать закономерность между множествами признаков X и Y. Характеристикой такого правила будут два параметра:

– достоверность (confidence) – определяет, какая часть объектов, содержащих признак X, содержит также и признак Y;

– поддержка (support) – определяет, какая часть объектов содержит оба признака.

Ассоциативные правила используют в этой статье для обнаружения вторжений. Использование НЛ преодолевает проблему резкой границы, имеющей место в случае их традиционного использования. Таким образом, полученные нечеткие ассоциативные правила могут быть использованы для поиска абстрактной корреляции среди различных характеристик безопасности. Интеграция НЛ с ассоциативными правилами и ГА позволяет генерировать более абстрактные и гибкие шаблоны для обнаружения вторжений, которые могут быть использованы как для обнаружения вредоносных действий, так и для выявления аномалий.

Подход, предложенный в [49], комбинирует ГА, НЛ и клеточный автомат (КА). Он может быть использован для классификации поведения программ на нормальное либо вредоносное. Неуправляемые КА, обученные на немаркированных данных, способны выявлять ранее неизвестные атаки. При этом учитывают как временную, так и пространственную информацию о сетевых соединениях, закодированную в правилах сетевой IDS. Авторы отмечают меньшее время обучения по сравнению с традиционными методами и прогнозируют эффективную работу по выявлению вредоносной активности с относительно низкими показателями ложных срабатываний.

В работе [50] описан подход, использующий МДР, а именно его реализацию C4.5. Основной идеей является применение петли обратной связи с участием специальных модулей, выполняющих роль эксперта по определенным видам атак и модулей, использующих метазнания. Авторы обосновывают возможность получения эффективных моделей и решений путем адаптации процессов машинного обучения и МИАД.

В работе [51] общая задача выявления аномалий несколько сужена до проведения интеллектуального анализа в отношении редких классов. Основная аргументация такого решения заключается в том, что доля классов, описывающих вторжения, много меньше классов, описывающих нормальное сетевое поведения. В статье приводят результаты исследования эффективности применения различных подходов, в частности, на основе МБС, коэффициента местного выброса (Local Outlier Factor – LOF), неуправляемого MOB (Unsupervised SVMs), расстояния Махаланобиса (Mahalanobis). Ключевым приоритетом проекта авторы заявляют способность алгоритма работать в режиме реального времени.

В работе [52] предлагают два метода для обучения детекторов аномалий: на основе обучающих правил (Learning Rules For Anomaly Detection – LERAD) и кластеризации (Clustering For Anomaly Detection – CLAD). Первый из них является обучающимся алгоритмом и может выразить нормальное поведение в логических правилах. Авторы отмечают относительно невысокий уровень ложных срабатываний. Второй метод не требует «чистых» данных. CLAD локализует аномалии путем поиска локальных и глобальных выбросов с некоторыми ограничениями. Используя алгоритмы k-NN и LOF, он допускает перекрытие кластеров и не стремится построить четкую модель. Поскольку CLAD не может точно описать состояние тревоги, авторы предлагают использовать иной термин – «возможность ошибки» (Near Miss). При анализе вторжений используют KDD99.

В работе [53] рассматривают основные принципы применения НЛ и ГА в системах обнаружения вторжений. Приводят решения, использующие систему нечеткого вывода совместно с ИНС, методику нейронного нечеткого обучения, методику нечеткого распознавания вторжений, комбинации ГА и техники нечетких МИАД, ГА, теории информации и т.д.

В другом исследовании [27] проводят сравнительный анализ результатов использования нечетких ГА для распознавания новых компьютерных атак с алгоритмами, использующими традиционный ГА и МДР. При тестировании использовались наборы данных KDD99 и RLD09. Необходимость использования RLD09 авторы со ссылкой на [54–55] объясняют устаревшими данными в KDD99 и отсутствием в них новых КА.

В работе [8] детально описываются принципы использования МДР для обнаружения вторжений и применения их к актуальным проблемам безопасности. При рассмотрении теории МДР затрагивается проблема поиска оптимального решения. Описаны принципы сбора исходных данных, формирования набора признаков для анализа и проверки точности конечного результата. Отдельный акцент делают на преимуществах этого подхода.

Еще один пример использования байесовского классификатора приведен в [56]. На этот раз он применяется в комбинации с ГА. Исходные данные классифицируют, после чего оценивают уровни выявленных атак и ложных срабатываний. Затем данные подвергают мутации и вновь классифицируют. На заключительном этапе проводят сравнительный анализ результатов классификации. Авторы отмечают существенное повышение количества выявленных угроз и снижение процента ложных срабатываний, а также большую эффективность предлагаемого метода по сравнению с решениями на базе ИНС, МОВ и традиционного байесовского классификатора.

В работе [57] проводят сравнительный анализ возможностей ИНС и МДР для решения задач выявления компьютерных атак. Исследователи приходят к выводу, что ИНС эффективны для обобщения и малопригодны для обнаружения новых атак, в то время как деревья решений эффективны для решения обеих задач. В то же время в работе [58] отмечают большой потенциал у методов, в основе которых лежат ИНС, в том числе и для решения задач обнаружения новых атак. В частности, отмечается алгоритм Snap-Drif в качестве наиболее перспективного кандидата.

В работе [59] исследуют вопрос применения МОВ и МДР для выявления попыток вторжений. Авторы приходят к выводу, что деревья решений дают лучшую точность, чем SVM, для атак типа зондирование, U2R, R2L. Для класса, соответствующего нормальной сетевой активности, точность у обоих методов примерно одинаковая, для класса DoS-атак точность выше у SVM.

4. Заключение

В работе рассмотрена проблематика применения МИАД для задач выявления компьютерных атак. Несмотря на очевидную привлекательность и перспективность этого направления, на сегодняшний день исследователи расходятся во мнении относительно эффективности того или иного подхода. Основной проблемой, ограничивающей применение МИАД, является относительно высокий уровень ложных срабатываний.

Неоднозначно отношение и к использованию KDD99 для отладки и оценки эффективности алгоритмов. С одной стороны KDD99 выступает в качестве своего рода общего знаменателя и обеспечивает некоторую объективность при сравнительном анализе различных подходов, с другой – использование устаревших данных для оценки вторжений вызывает определенные сомнения в общей достоверности полученных результатов. Для преодоления этой проблемы в отдельных случаях разрабатывают собственные наборы тестовых данных (например, RLD09).

В целом, наблюдается достаточно высокий уровень исследовательской активности в рассмотренной предметной области. Концентрация усилий в этом направлении, по всей видимости, объясняется глобальной значимостью вопросов, связанных с обеспечением ИБ, и со-

ответствующим уровнем технического развития, обеспечившим доступность необходимых вычислительных мощностей. Основной положительный эффект достигают за счет комбинации различных традиционных МИАД, что в ряде случаев как раз повышает эффективность анализа данных и обнаружения вторжений.

Литература

1. *Щавелев Л. В.* Способы аналитической обработки данных для поддержки принятия решений (СУБД.-1998 - №4-5). [Электронный ресурс], URL: http://infovisor.ivanovo.ru/press/paper04.html#28, (дата обращения: 08.01.2015).
2. *Маккафри Дж.* Кластеризация данных с использованием наивного байесовского вывода. [Электронный ресурс],
   URL: http://msdn.microsoft.com/ru-ru/magazine/jj991980.aspx, (дата обращения: 08.01.2015).
3. *Барсегян А. А., Куприянов М. С., Холод И. И., Тесс М. Д., Елизаров С. И.* Анализ данных и процессов: учеб. пособие. СПб : БХВ-Петербург, 2009. 512 с.
4. *Farid D. Md., Rahman M. Z., Rahman C. M.* Adaptive Intrusion Detection based on Boosting and Naïve Bayesian Classifier [Электронный ресурс].
   URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.3181&rep=rep1&type=pdf, (дата обращения: 16.02.2015).
5. *Brand E., Gerritsen R.* Naive-Bayes and Nearest Neighbor // DBMS. – 1998. – №7.
6. *Friedman N., Geiger D., Goldszmidt M.*, etc. Bayesian Network Classifiers // Machine Learning. 1997. № 29. С. 131–165.
7. *Heckerman D.* Bayesian Networks for Data Mining // Data Mining and Knowledge Discovery. 1997. № 1. С. 79–119.
8. *Markey J.* Using Decision Tree Analysis for Intrusion Detection: A How-To Guide. 2011 [Электронный ресурс].
   URL: http://www.sans.org/reading-room/whitepapers/detection/ decision-tree-analysis-intrusion-detection-how-to-guide-33678, (дата обращения: 27.01.2015).
9. *Шестаков К. М.* Курс лекций по специальному курсу «Теория принятия решений»: Электронная версия. Учебное пособие. [Электронный ресурс],
   URL: http://www.rfe.by/media/kafedry/kaf5/publikation/shestakov/teor-prinatia-resh-part1.doc, (дата обращения: 20.01.2015).
10. *Лифшиц Ю.* Метод опорных векторов. [Электронный ресурс],
    URL: http://logic.pdmi.ras.ru/~yura/internet/07ia.pdf , (дата обращения: 08.01.2015).
11. *Воронцов К. В.* Лекции по методу опорных векторов от 21 декабря 2007 года. [Электронный ресурс], URL: http://www.ccas.ru/voron/download/svm.pdf, (дата обращения: 08.01.2015).
12. Портал знаний. Глобальный интеллектуальный ресурс. [Электронный ресурс],
    URL: http://www.statistica.ru/branches-maths/metod-opornykh-vektorov-supported-vector-machine-svm, (дата обращения: 08.01.2015).
13. *Лепский А. Е., Броневич А. Г.* Математические методы распознавания образов. Курс лекций. [Электронный ресурс], URL: http://window.edu.ru/resource/800/73800/files/lect_Lepskiy_Bronevich_pass.pdf , (дата обращения: 27.03.2015).
14. *Воронцов К. В.* Лекции по метрическим алгоритмам классификации. [Электронный ресурс], URL: http://www.ccas.ru/voron/download/MetricAlgs.pdf, (дата обращения: 04.01.2015).

15. *Айвазян С. А., Бежаева З. И., Староверов О. В.* Классификация многомерных наблюдений. М.: Статистика, 1974. 240 с.
16. Технологии анализа данных. BaseGroup Labs. [Электронный ресурс], URL: http://www.basegroup.ru/library/analysis/regression/knn/, (дата обращения: 08.12.2015).
17. Tradeexperts. Математические основы k-nn. [Электронный ресурс], http://tradexperts.ru/ kNN_Osnovi.htm, (дата обращения: 04.01.2015).
18. Интуит. Национальный открытый университет. Лекция 9: Методы классификации и прогнозирования. Деревья решений. [Электронный ресурс], http://www.intuit.ru/studies/courses/6/6/lecture/174 (дата обращения: 10.12.2014).
19. База знаний кафедры ИКТ. МГИЭМ. Лекция3 - Методы построения деревьев решений. [Электронный ресурс], URL: http://wiki.auditory.ru/Лекция_3_-_Методы_построения_деревьев_решений, (дата обращения: 10.12.2014).
20. Анализ статистических данных с использованием деревьев решений [Электронный ресурс], URL: http://math.nsc.ru/AP/datamine/decisiontree.htm, (дата обращения: 15.12.2014 г.).
21. *Kumar S., Satbir J.* Intrusion Detection and Classification Using Improved ID3 Algorithm of Data Mining. [Электронный ресурс], URL: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-5-352-356.pdf, (дата обращения: 19.01.2015).
22. *Шампандар А. Дж.* Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия. М.: Вильямс, 2007. 768 с.
23. Universiteit Leiden. Leiden Institute of Advanced Computer Science. Decision Trees: an Introduction. [Электронный ресурс], URL: www.liacs.nl/~knobbe/intro_dec_tree.ppt, (дата обращения: 27.01.2015 г.).
24. *Николенко С.* Деревья принятия решений. Machine Learning CS Club, 2008. [Электронный ресурс], URL: http://logic.pdmi.ras.ru/~sergey/teaching/mlcsclub/02-dectrees.pdf, (дата обращения: 27.01.2015).
25. StatSoft. Электронный учебник по статистике. Деревья классификации. [Электронный ресурс], URL: http://www.statsoft.ru/home/textbook/modules/stclatre.html (дата обращения: 27.01.2015 г.).
26. *Lior R., Oded M.* Data mining with decision trees: Theory and Applications. World Scientific Publishing Co. Pte. Ltd, 2008. 244 с.
27. *Ireland E.* Intrusion Detection with Genetic Algorithms and Fuzzy Logic. [Электронный ресурс], URL: https://wiki.umn.edu/pub/UmmCSciSeniorSeminar/Fall2013PapersAndTalks/ cameraReadyCopy-EmmaIreland.pdf, (дата обращения: 08.01.2015).
28. *Круглов В. В., Дли М. И., Голунов Р. Ю.* Нечеткая логика и искусственные нейронные сети. М.: Физматлит, 2001. 224 с.
29. *Воронцов К. В.* Лекции по искусственным нейронным сетям от 21 декабря 2007 г. [Электронный ресурс], URL: http://www.ccas.ru/voron/download/NewralNetworks.pdf, (дата обращения: 03.02.2015).
30. *Барский А. Б.* Нейронные сети: распознавание, управление, принятие решений. М. : Финансы и статистика, 2004. 176 с.
31. *Беркинблит М. Б.* Нейронные сети: Учебное пособие. М.: МИРОС, 1993. 96 с.
32. *Каллан Р.* Основные концепции нейронных сетей: Пер. с англ. М.: Вильямс, 2001. 287 с.
33. *Лю Б.* Теория и практика неопределенного программирования. М.: БИНОМ, 2005. 416 с.
34. *Галушкин А. И.* Нейрокомпьютеры и их применение. М.: ИПРЖР, 2000. 416 с.
35. *Хайкин С.* Нейронные сети: полный курс. М. : Издательский дом "Вильямс", 2006. 1104 с.

36. *Киселев М., Соломатин, Е.* Средства добычи знаний в бизнесе и финансах // Открытые системы. 1997. № 4. С. 41–44.
37. *Mumick I. S., Quass D., Mumick B. S.* Maintenance of Data Cubes and Summary Tables in a Warehouse. Stanford University, Database Group, 1996 [Электронный ресурс]. URL: http://infolab.stanford.edu/pub/papers/cube-maint.ps, (дата обращения: 27.03.2015).
38. *Шлаев Д. В.* Нечеткая логика - математические основы. [Электронный ресурс], URL: http://www.stgau.ru/company/personal/user/8068/files/lib/Очная форма обучения/Интеллектуальные информационные системы/Лекции/6.1_Нечеткая логика.pdf (дата обращения: 30.01.2015).
39. *Шеври Ф., Гели Ф.* Electric. Выпуск No 31.Нечеткая логика. 2009г. [Электронный ресурс], URL: http://www.netkom.by/docs/N31-Nechetkaya-logika.pdf, (дата обращения: 20.01.2015).
40. *Штовба С. Д.* Введение в теорию нечетких множеств и нечеткую логику. [Электронный ресурс], http://matlab.exponenta.ru/fuzzylogic/book1/ (дата обращения: 20.01.2015).
41. BaseGroup Labs: технологии анализа данных. Нечеткая логика - математические основы. [Электронный ресурс],
URL: http://www.basegroup.ru/library/analysis/fuzzylogic/math (дата обращения: 20.01.2015).
42. *Головицына М.* Интуит. Национальный открытый университет. Информационные технологии в экономике: Информация. [Электронный ресурс], URL: http://www.intuit.ru/studies/courses/3735/977/lecture/14689, (дата обращения: 22.01.2015).
43. *Панченко Т. В.* Генетические алгоритмы: учебно-методическое пособие / под ред. Тарасевича Ю. Ю. Астрахань : Издательский дом «Астраханский университет», 2007. 87 с.
44. *Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А.* Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Х.: ОСНОВА, 1997. 112 с.
45. *Рутковская Д., Пилиньский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия – Телеком, 2006. 452 с.
46. KDD Cup 1999 Data [Электронный ресурс]. URL: http://kdd.ics.edu/databases/kddcup99/ kddcup99.html, (дата обращения: 08.01.2015).
47. *Farid D. Md., Harbi N., Rahman M. Z.* Combining naive bayes and decision tree for adaptive intrusion detection. // International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010 [Электронный ресурс]. URL: http://arxiv.org/pdf/1005.4496v1.pdf.
48. *Harshna, Kaur N.* Fuzzy Data Mining Based Intrusion Detection System Using Genetic Algorithm. January 2014 [Электронный ресурс].
URL: http://www.ijarcce.com/upload/2014/ january/ IJARCCE3I__a_harshna_fuzzy.pdf, (дата обращения: 16.02.2015).
49. *P. Kiran Sree, I. Ramesh Babu.* Investigating Cellular Automata Based Network Intrusion Detection System For Fixed Networks (NIDWCA) [Электронный ресурс]. URL: http://arxiv.org/pdf/1401.3046.pdf, (дата обращения: 16.02.2015).
50. *Adebowale A.* An Enhanced Data Mining Based Intrusion Detection System (IDS) using Selective Feedback. September 2013 [Электронный ресурс]. URL: http://ijcit.com/archives/ volume2/issue5/Paper020535.pdf, (дата обращения: 16.02.2015).
51. *Dokas P, Ertoz L., Kumar V.* et al. Data Mining for Network Intrusion Detection [Электронный ресурс]. URL: http://minds.cs.umn.edu/papers/nsf_ngdm_2002.pdf.

52. *Chan P. K., Mahoney M. V., Arshad M. H.* Learning rules and clusters for anomaly detection in network traffic [Электронный ресурс].
URL: http://citeseerx.ist.psu.edu/viewdoc/ download?doi=10.1.1.71.9357&rep=rep1&type=pdf.
53. *Borgohain R.* FuGeIDS: Fuzzy Genetic paradigms in Intrusion Detection Systems [Электронный ресурс].
URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.302.7753.
54. *Jongsuebsuk P., Wattanapongsakorn N., Charnsripinyo C.* Network intrusion detection with Fuzzy Genetic Algorithm for unknown attacks. // In Information Networking (ICOIN), 2013 International Conference. 2013. P. 1–5.
55. *Jongsuebsuk P., Wattanapongsakorn N., Charnsripinyo C.* Real-time intrusion detection with fuzzy genetic algorithm. // In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference. 2013. P. 1–6.
56. *Y. V. Srinivasa Murthy, Harish K., D. K. Vishal Varma, Sriram K.* et al. Hybrid Intelligent Intrusion Detection System using Bayesian and Genetic Algorithm (BAGA): Comparitive Study [Электронный ресурс]. URL: http://research.ijcaonline.org/volume99/number2/pxc3897808.pdf, (дата обращения: 10.02.2015).
57. *Bouzida Y., Cuppens F.* Neural networks vs. decision trees for intrusion detection [Электронный ресурс].
URL: http://www.telecom-bretagne.eu/data/publications/ ArticlesConference/monam06.pdf, (дата обращения: 11.02.2015).
58. *Beqiri E.* Neural Networks for Intrusion Detection Systems [Электронный ресурс]. URL: http://www.freepapers.ir/PDF/10.1007-978-3-642-04062-7_17.pdf?hash=2aKa8wNG5cWd3I95vZFm1g, (дата обращения: 05.02.2015).
59. *Peddabachigari S., Abraham A., Thomas J.* Intrusion Detection Systems Using Decision Trees and Support Vector Machines [Электронный ресурс].
URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.4079&rep=rep1&type=pdf, (дата обращения: 17.02.2015).

Статья поступила в редакцию 11.01. 2016

Зубков Евгений Валерьевич
аспирант кафедры безопасности и управления в телекоммуникациях СибГУТИ, тел.+7-913-798-01-48, e-mail: evz.nsk@gmail.com

Белов Виктор Матвеевич
д.т.н., профессор, профессор кафедры безопасности и управления в телекоммуникациях СибГУТИ, тел.+7-963-906-84-83, e-mail: vmbelov@mail.ru

Data mining and intrusion detection methods

E. Zubkov, V. Belov

The article focuses on the problems of data mining techniques used for intrusion detection. This question is a key factor in building intrusion detection systems (IDS) including anomaly-based network IDS (NIDS). The article provides an overview of the most popular approaches used for intrusion detection and examples of applications.

Keywords: data mining, information security, intrusion detection, anomaly, naive bayes approach, support vector machine, nearest neighbor, decision tree, artificial neural networks, fuzzy logic, genetic algorithms.