УДК 004.056 **DOI** 10.15622/sp.52.10

А.А. Браницкий

ИЕРАРХИЧЕСКАЯ ГИБРИДИЗАЦИЯ БИНАРНЫХ КЛАССИФИКАТОРОВ ДЛЯ ВЫЯВЛЕНИЯ АНОМАЛЬНЫХ СЕТЕВЫХ СОЕДИНЕНИЙ

Браницкий А.А. **Иерархическая гибридизация бинарных классификаторов для выявления аномальных сетевых соединений.**

Аннотация. В статье предлагается обобщенный гибридный подход к построению коллектива классификационных правил на примере решения задачи выявления аномальных сетевых соединений. Выделяется пять этапов в рассматриваемой методике. Первый этап включает в себя настройку адаптивных классификаторов. На втором этапе выполняется сигнатурный анализ, сборка сетевых соединений и формирование сетевых параметров. Третий этап заключается в предобработке сетевых параметров. На четвертом этапе осуществляется обход в ширину дерева классификаторов совместно с их обучением или тестированием. На пятом этапе выявляются аномальные сетевые соединения. Особенностями предлагаемой методики являются возможность задания произвольной вложенности классификаторов друг в друга и ленивое подключение классификаторов благодаря нисходящему каскадному обучению общего коллектива классификационных правил. Приводятся результаты экспериментов с использованием открытого набора данных для вычисления показателей эффективности обнаружения и классификации сетевых аномалий.

Ключевые слова: сетевые аномалии, сетевые соединения, протоколы TCP/IP, гибридизация классификаторов.

1. Введение. Развитие современных технологий способствует росту сетевого трафика, передаваемого с использованием протоколов семейства TCP/IP [1]. Для обеспечения безопасности и повышенной отказоустойчивости оконечных и коммутирующих сетевых узлов необходимо применять специальные программные средства — системы обнаружения и предотвращения сетевых атак. Основной их целью является контроль и анализ захваченных пакетов на предмет аномального содержимого на различных уровнях стека протоколов. Поскольку данные в сети Интернет передаются обособленными связанными блоками (фрагментами), анализ на уровне отдельных пакетов является недостаточным для выявления большинства сетевых аномалий, направленных на захват или выведение из строя вычислительного узла. Среди таких аномалий можно назвать вирусную деятельность, перегрузки сетевого оборудования, атаки типа «отказ в обслуживании», сканирование портов и хостов. Как правило, для обнаружения подобного рода атак требуется гораздо большее число пакетов, объединенных в минимальный сетевой поток — соединение, признаки которого могут служить в качестве входных параметров для настройки адаптивных моделей. В данной статье в качестве таких моделей рассмотрены многослойная нейронная сеть, нейронечеткая сеть на основе вывода

Такаги — Сугено и машина опорных векторов. Также для повышения качественных характеристик отдельных моделей предлагается использовать несколько коллективов решателей, а именно мажоритарное голосование, многоярусную укладку и объединение с использованием арбитра на основе динамических областей компетентности.

Данная статья продолжает развитие цикла работ, посвященных анализу защищенности и реагированию на атаки в компьютерных сетях, а также обнаружению целевых атак в распределенных крупномасштабных критически важных системах [2], и базируется на предыдущих работах автора [3, 4]. Научный вклад настоящей статьи состоит в представлении обобщенной методики, позволяющей выполнять объединение разнородных классификаторов, то есть процедуру гибридизации, с приложением к области обнаружения аномальных сетевых соединений. Отметим, что такой подход может быть использован и вне этой области для решения более общих задач классификации объектов.

Статья имеет следующую структуру. Первый раздел — введение. Второй раздел включает в себя постановку задачи исследования и обзор некоторых работ, связанных с обнаружением сетевых атак при помощи комбинированных подходов, базирующихся на адаптивных классификаторах. В третьем, четвертом и пятом разделах рассматриваются модели бинарных классификаторов, которые используются в данном исследовании в качестве минимальных блоков при обнаружении и классификации сетевых аномалий. В шестом разделе представлена методика иерархической гибридизации бинарных классификаторов для выявления аномальных сетевых соединений. Седьмой раздел содержит экспериментальную оценку предложенной методики с использованием открытого набора сырых сетевых дампов. Восьмой раздел — заключение.

2. Постановка задачи и релевантные работы. Задача выявления аномальных сетевых соединений при помощи объединения классификаторов может быть сформулирована следующим образом. Даны базовые классификаторы $F^{(1)},\ldots,F^{(s)}:\mathbb{R}^n \to 2^{\{0,\ldots,m\}}$, обученные на наборе маркированных векторов признаков сетевых соединений $\chi=\{(\boldsymbol{X_i},c_i)\}_{i=1}^{M}$ ($c_i\in\{0,\ldots,m\}$), и их агрегирующая композиция (функция коллектива классификационных правил) $F:\{0,\ldots,m\}^s\times\mathbb{R}^n \to 2^{\{0,\ldots,m\}}$, которая комбинирует выходные результаты классификаторов $F^{(1)},\ldots,F^{(s)}$. Каждое из представленных классификационных правил $F^{(1)},\ldots,F^{(s)},F$ в качестве выходного значения формирует множество $\{c_i'\}_{i=0}^{m'}\subset\{0,\ldots,m\}$ ($0\leqslant m'\leqslant m$), чьи эле-

менты обозначают возможные метки классов с точки зрения этого классификатора. Кроме того, сама функция F может представлять собой сложную многоуровневую процедуру, что затрудняет разработку общего подхода для построения коллектива решателей. Требуется произвести настройку функции F таким образом, чтобы функционал эмпирического риска ее композиции с базовыми классификаторами

$$\psi_\chi\left(F\circ[F^{(1)},\ldots,F^{(s)},\mathrm{id}]\right)=\frac{1}{M}\cdot\#\left\{\boldsymbol{X_i}\mid F\left(F^{(1)}(\boldsymbol{X_i}),\ldots,F^{(s)}(\boldsymbol{X_i}),\boldsymbol{X_i}\right)\neq\{c_i\}\right\}_{i=1}^{M}$$

не превышал среднего арифметического функционалов эмпирического риска отдельных классификационных правил $F^{(1)},\ldots,F^{(s)}$:

$\psi_\chi\left(F\circ[F^{(1)},\ldots,F^{(s)},\mathrm{id}]\right)\leqslant\frac{1}{s}\cdot\sum_{j=1}^{s}\psi_\chi\left(F^{(j)}\right)$. Используемое в правой части суммарное усреднение может быть заменено $\min_{j\in\{1,\ldots,s\}}\psi_\chi\left(F^{(j)}\right)$. На рисунке 1 представлена одна из возможных схем объединения бинарных классификаторов при помощи функций $F^{(1)},\ldots,F^{(s)},F$, более подробное разъяснение которой приведено в разделе 6.

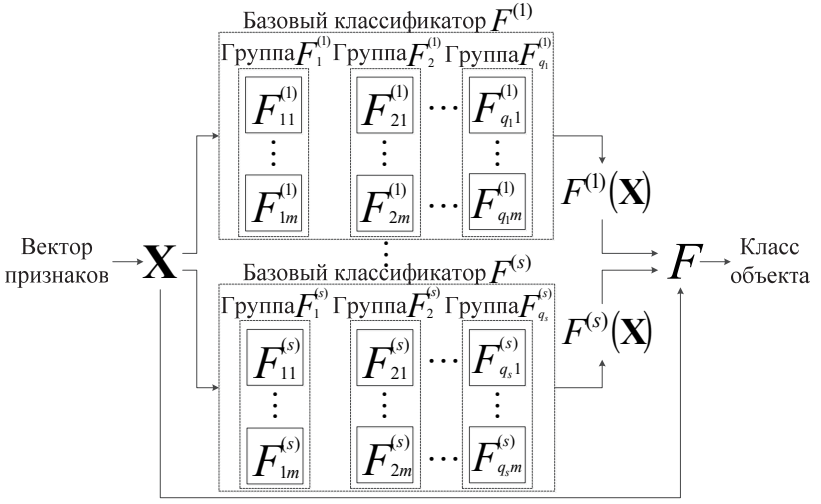

Рис. 1. Предлагаемый подход объединения бинарных классификаторов

Обнаружение аномальных сетевых соединений при помощи адаптивных классификаторов является активно исследуемой областью. Для решения этой задачи в [5] предлагается использовать K нейронных сетей с радиальными базисными функциями, причем каждая из

этих сетей обучается на различных дизъюнктных подмножествах $D_1,\ldots,D_K$ исходного обучающего набора данных D. Такие подмножества генерируются при помощи метода нечеткой кластеризации, согласно которому каждый элемент $\boldsymbol{X} \in D$ относится к области D_i с некоторой степенью принадлежности $u_i^{\boldsymbol{X}}$. Каждое подмножество D_i, $(i=1,\ldots,K)$, составляется из тех элементов, которые имеют наибольшую степень принадлежности к этому подмножеству среди всех остальных подмножеств. По словам авторов, за счет такого предварительного разбиения улучшается обобщающая способность классификаторов и сокращается время их обучения, поскольку для их настройки используются только те объекты, которые наиболее плотно сгруппированы вокруг образовавшегося центра обучающего подмножества. Для объединения выходных результатов $\boldsymbol{Y_1},\ldots,\boldsymbol{Y_K}$ этих классификаторов, принимающих на входе вектор $\boldsymbol{X}$, используется многослойная нейронная сеть, входной вектор для которой представляется в виде набора элементов, полученных в результате применения пороговой функции к каждому компоненту вектора $u_i^{\boldsymbol{X}} \cdot \boldsymbol{Y_i}$ $(i=1,\ldots,K)$. Аналогичный подход был использован ранее в [6], где в роли базовых классификаторов выступали нейронные сети прямого распространения, а входом для агрегирующего их модуля, представленного также в виде классификатора указанного типа, являлись непосредственно значения векторов $u_1^{\boldsymbol{X}} \cdot \boldsymbol{Y_1},\ldots,u_K^{\boldsymbol{X}} \cdot \boldsymbol{Y_K}$.

В [7] для анализа записей о сетевых соединениях используются нейронечеткие модели и машины опорных векторов. Авторы этой работы выделяют четыре основных этапа в предлагаемом ими подходе. На первом этапе осуществляется генерация обучающих данных при помощи метода k-средних. Второй этап — обучение нейронечетких классификаторов. На третьем этапе выполняется формирование входного вектора для машины опорных векторов. Заключительный этап — обнаружение атаки при помощи последнего классификатора.

В [8] для обнаружения каждого из трех типов DDoS-атак, осуществляемых с использованием протоколов TCP, UDP и ICMP, строится отдельная нейронная сеть с одним скрытым слоем. Последний слой каждой такой нейронной сети состоит из одного узла, выходное значение которого интерпретируется как наличие или отсутствие DDoS-атаки соответствующего типа. Предлагаемый подход реализован как модуль в системе обнаружения атак Snort и протестирован на трафике реального сетевого окружения.

В [9] для обнаружения DoS-атак предлагается использовать комбинированный подход, совмещающий в себе метод нормализованной энтропии и машины опорных векторов. Для выявления аномалий из сетевого трафика извлекаются шесть показателей, численно выраженных как интенсивность появления различных значений выбранных полей внутри пакетов в рамках 60-секундного окна. В данном подходе сперва вычисляются сетевые параметры при помощи метода нормализованной энтропии, затем они используются в качестве входных обучающих и тестовых данных для машины опорных векторов.

В [10] для обнаружения DoS-атак и сканирования хостов рассматривается подход, основанный на последовательном применении процедуры сжатия векторов признаков сетевых соединений и двух нечетких преобразований. Сперва ко входному восьмимерному вектору применяется метод главных компонент, уменьшающий размерность аргумента до пяти компонент с сохранением относительной суммарной дисперсии на уровне более 90%. Следующий шаг заключается в обучении или тестировании нейронечеткой сети, выходное значение которой обрабатывается при помощи метода нечеткой кластеризации.

3. Модель нейронной сети для выявления аномальных сетевых соединений. В искусственной *нейронной сети* моделирование наличия и силы входных импульсов между ее вычислительными элементами — нейронами — может быть установлено через задание ненулевых весовых коэффициентов соответствующих связей. После настройки подобные структуры, задаваемые как минимум двумя слоями, способны выполнять достаточно точную аппроксимацию элементов обучающей выборки [11-14].

Входной слой нейронной сети представляет собой фиктивный слой, который выполняет функцию предварительного распределения поступающих сигналов перед их непосредственной обработкой. Входной вектор каждого узла первого скрытого слоя — это скалярное произведение вектора синаптических весов и входного вектора $\boldsymbol{X} = (x_1, \ldots, x_n)^T$. Сигнал, поступающий на вход i'-ого нейрона первого скрытого слоя, состоящего из N_1 узлов, конструируется следующим образом: $x_{i'}^{(1)} = \sum_{j=1}^{n} w_{i'j}^{(1)} \cdot x_j + \theta_{i'}^{(1)}$, где $i' = 1, \ldots, N_1$, $w_{i'j}^{(1)}$ — веса, задающие преобразование сигналов $\boldsymbol{X}$ на входе i'-ого нейрона первого скрытого слоя, $\theta_{i'}^{(1)}$ — параметр смещения i'-ого нейрона, размещенного в первом скрытом слое. Выходным сигналом рассматриваемого нейрона можно считать величину $y_{i'}^{(1)} = \varphi\left(x_{i'}^{(1)}\right)$. Аналогичным

образом задаются входной и выходной сигналы для каждого i''-ого нейрона, который расположен во втором скрытом слое, имеющем N_2 нейронов: $x_{i''}^{(2)} = \sum_{j=1}^{N_1} w_{i''j}^{(2)} \cdot y_j^{(1)} + \theta_{i''}^{(2)}$ и $y_{i''}^{(2)} = \varphi\left(x_{i''}^{(2)}\right)$, где $i'' = 1,\ldots,N_2$, $w_{i''j}^{(2)}$ — веса, задающие преобразование сигналов $\boldsymbol{Y^{(1)}} = (y_1^{(1)},\ldots,y_{N_1}^{(1)})^T$ на входе i''-ого нейрона, размещенного во втором скрытом слое, $\theta_{i''}^{(2)}$ — параметр смещения i''-ого нейрона, φ — функция активации. Результирующий сигнал Y составляется следующим образом: $y_1^{(3)} = \varphi\left(\sum_{j=1}^{N_2} w_{1j}^{(3)} \cdot y_j^{(2)} + \theta_1^{(3)}\right)$, где $w_{1j}^{(3)}$ — веса на входе нейрона последнего слоя, $\theta_1^{(3)}$ — параметр смещения выходного нейрона.

Таким образом, функционирование модели одноклассовой нейронной сети может быть описано следующей формулой:

$$Y(\boldsymbol{X}) = \varphi\left(\sum_{i=1}^{N_2} w_{1i}^{(3)} \cdot \varphi\left(\sum_{j=1}^{N_1} w_{ij}^{(2)} \cdot \varphi\left(\sum_{k=1}^{n} w_{jk}^{(1)} \cdot x_k + \theta_j^{(1)}\right) + \theta_i^{(2)}\right) + \theta_1^{(3)}\right).$$

Рассмотрим алгоритм обратного распространения ошибки, который является наиболее распространенным алгоритмом обучения многослойных нейронных сетей: (1) задание структуры нейронной сети (выбор числа скрытых слоев и нейронов, расположенных в них); (2) инициализация весовых коэффициентов $w_{ij}^{(K)}$ произвольными значениями, где K обозначает номер слоя, i соответствует номеру позиции нейрона в K-ом слое, j отображает наличие связи между текущим нейроном и выходным сигналом j-ого нейрона в $(K-1)$-ом слое; (3) задание максимального числа итераций обучения (эпох) T и минимального значения суммарной среднеквадратичной ошибки ε; (4) прямое распространение сигналов: вычисление входящих сигналов для каждого i-ого нейрона в K-ом слое по формуле $x_i^{(K)} = \sum_{j=1}^{N_{K-1}+1} w_{ij}^{(K)} \cdot y_j^{(K-1)}$, где N_{K-1} — число нейронов в $(K-1)$-ом слое, $w_{ij}^{(K)} = \theta_i^{(K)}$ и $y_j^{(K-1)} = 1$ для $j = N_{K-1}+1$, $y_j^{(K-1)} = \varphi\left(x_j^{(K-1)}\right)$ для $K > 1$ и $y_j^{(K-1)} = x_j$ (исходный сигнал) для $K = 1$; (5) обратное распространение ошибки: вычисление приращений весовых коэффициентов нейронов по формуле: $\Delta w_{ij}^{(K)} = \alpha \cdot \delta_i^{(K)} \cdot y_j^{(K-1)}$, последовательно начи-

ная с последнего слоя и заканчивая первым ($0 < \alpha \leqslant 1$ — коэффициент пропорциональности коррекции весов). Если K-ый слой выходной, то $\delta_i^{(K)} = \varphi'\left(x_i^{(K)}\right) \cdot \left(r_i - y_i^{(K)}\right)$, иначе $\delta_i^{(K)} = \varphi'\left(x_i^{(K)}\right) \cdot \sum_{j=1}^{N_{K+1}+1} \delta_j^{(K+1)} \cdot w_{ji}^{(K+1)}$, где r_i обозначает желаемый выход нейронной сети в i-ом нейроне на выходном слое; (6) корректировка весовых коэффициентов нейронов по формуле: $w_{ij}^{(K)}(t+1) = w_{ij}^{(K)}(t) + \Delta w_{ij}^{(K)}$, где t обозначает номер итерации алгоритма; (7) останов алгоритма при выполнении одного из условий: $t > T$ или $\sum_{\boldsymbol{X} \in \{\boldsymbol{X_i}\}_{i=1}^{M}} E(\boldsymbol{X}) \leqslant \varepsilon$, где $E(\boldsymbol{X}) = \frac{1}{2} \cdot \sum_{i=1}^{N_{Kall}} \left(r_i - y_i^{(K_{all})}\right)^2$ — среднеквадратичная ошибка нейронной сети, имеющей K_{all} слоев и $N_{K_{all}}$ нейронов на выходном слое, при подаче вектора $\boldsymbol{X}$ на ее распределительный слой; в противном случае переход к шагу 4.

Приведенный выше алгоритм принадлежит к общему семейству алгоритмов градиентного спуска, в которых поиск точки минимума осуществляется в направлении, противоположном градиенту оптимизируемой функции (например, среднеквадратичной ошибки). Для таких алгоритмов характерно «проваливание в яму локального минимума», когда алгоритм практически прекращает модифицировать весовые параметры, несмотря на наличие более глубокого экстремума по сравнению с уже найденным. Эти проблемы частично решаются при помощи различных улучшений алгоритма обратного распространения ошибки, которые могут использовать переменный коэффициент пропорциональности коррекции весов в зависимости от сохранения/изменения знака производной [15] или принимать во внимание факторы момента для изменения каждого отдельного веса [16].

4. Модель нейронечеткой сети для выявления аномальных сетевых соединений. Следующий подход, используемый при построении интеллектуального ядра для выявления сетевых аномалий — это *нейронечеткие сети*, являющиеся частным случаем систем нечеткого вывода, которые отражают способность человеческого мышления принимать решения в условиях неопределенности и нечеткости. Как правило, такие системы состоят из пяти функциональных блоков [17]. Первый блок — это база правил, которая включает набор нечетких импликаций (правил) вида if A then B. Левая часть A такого правила называется посылкой, правая часть B — заключением. Такие правила существенно отличаются от традиционных продукционных тем, что каждому из утверждений, входящих в состав частей A и B, приписывается некоторое число от 0 до 1, отражающее степень достоверности

посылки и заключения. Второй блок — это база данных, содержащая набор функций принадлежности. Эти функции задают для входных лингвистических переменных переход от их количественных (crisp) значений к нечетким лингвистическим термам. Для каждого из таких термов строится отдельная функция принадлежности, выходное значение которой характеризует меру принадлежности входной переменной соответствующему нечеткому множеству (терму). Наиболее часто используемым типом функций принадлежности являются непрерывные кусочно-дифференцируемые (треугольные и трапецеидальные функции) или гладкие функции (семейство колоколообразных функций) с областью значений [0, 1]. Третий блок — блок фаззификации (введения нечеткости), роль которого заключается в применении к входному аргументу заданной функции принадлежности соответствующего ей лингвистического терма. Каждый из конъюнктов A_i, входящих в состав посылки $A = A_1 \wedge \ldots \wedge A_n$, и заключение B представляются в виде нечетких утверждений x_i is γ_i и y is Γ соответственно, где x_i и y — лингвистические переменные, γ_i и Γ — лингвистические термы. Результатом этапа фаззификации является набор вычисленных значений этих нечетких утверждений. Четвертый блок — блок нечеткого вывода, содержащий набор уже встроенных в его ядро нечетких импликаций и предоставляющий механизм (к примеру, правило modus ponens или modus tollens) для вычисления заключения B по входному набору конъюнкций в части посылки A. Для вычисления полной степени истинности левой части применяются Т-нормы, наиболее распространенными примерами которых являются операции минимума и произведения. На выходе блока нечеткого вывода для лингвистической переменной y формируется один или несколько нечетких термов вместе с соответствующими для них значениями функций принадлежности. Пятый блок — блок дефаззификации (приведения к четкости), восстанавливающий количественное значение лингвистической переменной y по ее нечетким значениям. А именно: полученные в результате работы блока нечеткого вывода данные преобразуются в количественные значения при помощи одного из следующих методов: метода центра площади, метода центра тяжести, метода суммы центров, метода максимума функции принадлежности.

В описанной системе нечеткого вывода заключения B во всех правилах if A then B имели вид нечеткого утверждения y is Γ, которое не зависит от лингвистических переменных, входящих в состав посылки A. Подход, предложенный Такаги и Сугено [18], направлен

на устранение этого недостатка и заключается во введении в правую часть каждого из правил некоторой функциональной зависимости от элементов его левой части, а именно $y = f(x_1,\ldots,x_n)$. В ситуациях, приближенных к реальным жизненным, часто приходится сталкиваться с моделями подобного типа, в частности, когда человек или устройство не имеет возможности точно оценить величины входных параметров, но при этом регулирующее воздействие может быть явно вычислено по известной формуле.

Нейронечеткая сеть (ANFIS) [17] является развитием модели Такаги — Сугено [18], в которую добавлен элемент адаптивной настройки (обучения) ее параметров. Такая сеть состоит из пяти слоев, где входные сигналы претерпевают изменения, распространяясь последовательно от первого до последнего слоя. Каждое нечеткое правило в сети представляется как элемент, принадлежащий набору правил вида:

$$\left\{\text{if}\,(x_1 \text{ is}\,\gamma_1^{(j1)} \wedge \ldots \wedge x_n \text{ is}\,\gamma_n^{(jn)})\ \text{then}\, y = f^{(j)}(x_1,\ldots,x_n) = p_0^{(j)} + p_1^{(j)}\cdot x_1 + \ldots + p_n^{(j)}\cdot x_n\right\}_{j=1}^{P}.$$

Здесь P обозначает мощность набора нечетких правил, в которых каждая переменная $x_1,\ldots,x_n$ имеет ровно m нечетких термов; $j_1,\ldots,j_n$ обозначают номера нечетких термов, соответствующих лингвистическим переменным $x_1,\ldots,x_n$, в нечетком правиле под номером j $(1\leqslant j_1\leqslant m,\ldots,1\leqslant j_n\leqslant m)$. Как и в классической системе нечеткого вывода, левая часть такого правила является конъюнкцией нечетких утверждений, которые выражают степень соответствия входного количественного значения x_i тому или иному лингвистическому терму $\gamma_i^{(j_i)}$ согласно выражению $\mu_{\gamma_i^{(j_i)}}(x_i)$, где в качестве функции принадлежности $\mu_{\gamma_i^{(j_i)}}$ чаще всего используются колоколообразная функция $\left(1+|(x-c_{ij})/a_{ij}|^{2\cdot b_{ij}}\right)^{-1}$ или гауссова функция $\exp\left\{-((x-c_{ij})/a_{ij})^2\right\}$, где $i=1,\ldots,n$ и $j=1,\ldots,P$. Узловые элементы первого слоя в нейронечеткой сети выполняют роль фаззификации входной лингвистической переменной x_i, и выходом этого слоя являются значения функции принадлежности $\mu_{\gamma_i^{(j)}}$ этой переменной нечеткому множеству (терму) $\gamma_i^{(j)}$: $Y_{ji}^{(1)} = \mu_{\gamma_i^{(j)}}(x_i)$, где $i=1,\ldots,n$, $j=1,\ldots,m$. Во втором слое осуществляется формирование посылок нечетких правил с их объедине-

нием при помощи операции взятия Т-нормы — произведения; выход этого слоя можно рассматривать как вес правила: $Y_k^{(2)} = Y_{k_1 1}^{(1)} \times \ldots \times Y_{k_n n}^{(1)} = \mu_{\gamma_1^{(k_1)}}(x_1) \times \ldots \times \mu_{\gamma_n^{(k_n)}}(x_n)$, где $k = 1, \ldots, P$, причем $P \leqslant m^n$ при условии отсутствия каких-либо противоречащих друг другу правил. В элементах третьего слоя вычисляется отношение веса соответствующего правила к общей сумме весов всех правил, выходом этого слоя является нормализованная к [0, 1] величина: $Y_k^{(3)} = Y_k^{(2)} \Big/ \sum_{i=1}^{P} Y_i^{(2)}$. В четвертом слое вычисляется результат заключения каждого из правил с учетом полученной на третьем слое относительной степени его выполнения; выход этого слоя отражает аддитивную долю каждого правила в общем выходе сети: $Y_k^{(4)} = Y_k^{(3)} \cdot f^{(k)}(x_1, \ldots, x_n) = Y_k^{(3)} \cdot \left(p_0^{(k)} + p_1^{(k)} \cdot x_1 + \ldots + p_n^{(k)} \cdot x_n \right)$. На выходном пятом слое располагается единственный нейрон, отвечающий за суммирование входных сигналов, поступающих от узлов четвертого слоя: $Y^{(5)} = \sum_{i=1}^{P} Y_i^{(4)} = \sum_{i=1}^{P} Y_i^{(3)} \cdot f^{(i)}(x_1, \ldots, x_n) = \dfrac{\sum_{i=1}^{P} Y_i^{(2)} \cdot f^{(i)}(x_1, \ldots, x_n)}{\sum_{i=1}^{P} Y_i^{(2)}}$.

Тем самым модель ANFIS представляется при помощи следующего соотношения:

$$Y(\boldsymbol{X}) = \frac{\sum_{i=1}^{P} \left(\mu_{\gamma_1^{(i_1)}}(x_1) \times \ldots \times \mu_{\gamma_n^{(i_n)}}(x_n) \right) \cdot \left(p_0^{(i)} + p_1^{(i)} \cdot x_1 + \ldots + p_n^{(i)} \cdot x_n \right)}{\sum_{i=1}^{P} \mu_{\gamma_1^{(i_1)}}(x_1) \times \ldots \times \mu_{\gamma_n^{(i_n)}}(x_n)}.$$

Рассмотрим алгоритм обучения нейронечеткой сети, построенной на основе системы нечеткого вывода Такаги — Сугено: (1) задание множества лингвистических термов $\left\{ \gamma_i^{(j)} \right\}_{j=1}^{m}$ для каждой из входных лингвистических переменных x_i ($i = 1, \ldots, n$); (2) выбор типа функций принадлежности $\mu_{\gamma_i^{(j)}}$ для каждого лингвистического терма $\gamma_i^{(j)}$; (3) задание максимального числа итераций обучения (эпох) T и минимального значения суммарной среднеквадратичной ошибки ε; (4) вычисление суммарных квадратов расхождений между каждым компонентом желаемого выходного вектора $\boldsymbol{R}_{\boldsymbol{X}} = (r_{\boldsymbol{X},1}, \ldots, r_{\boldsymbol{X},N_K})^T$ и каждым компонентом фак-

тического выходного вектора $\boldsymbol{Y}_{\boldsymbol{X}}^{(\boldsymbol{K})} = (y_{\boldsymbol{X},1}^{(K)},\ldots,y_{\boldsymbol{X},N_K}^{(K)})^T$ нейронечеткой сети для каждого подаваемого на ее вход обучающего вектора $\boldsymbol{X} = (x_1,\ldots,x_n)^T$: $E_{\boldsymbol{X}} = \frac{1}{2}\cdot\sum_{i=1}^{N_K}(r_{\boldsymbol{X},i} - y_{\boldsymbol{X},i}^{(K)})^2$, где $K = 5$ — общее количество слоев в сети, $N_K = 1$ — размерность выходного слоя; (5) вычисление уровня ошибок (направления, в котором происходит убывание функции $E_{\boldsymbol{X}}$) для выходного слоя сети: $\frac{\partial E_{\boldsymbol{X}}}{\partial y_{\boldsymbol{X},i}^{(K)}} = -(r_{\boldsymbol{X},i} - y_{\boldsymbol{X},i}^{(K)})$, где $i = 1,\ldots,N_K$; (6) вычисление уровня ошибок для внутренних слоев сети: $\frac{\partial E_{\boldsymbol{X}}}{\partial y_{\boldsymbol{X},i}^{(L)}} = \sum_{j=1}^{N_L}\frac{\partial E_{\boldsymbol{X}}}{\partial y_{\boldsymbol{X},j}^{(L+1)}}\cdot\frac{\partial y_{\boldsymbol{X},j}^{(L+1)}}{\partial y_{\boldsymbol{X},i}^{(L)}}$, где $i = 1,\ldots,N_L$, $L < K$ и уровня ошибок для заданных в их узлах параметров τ сети: $\frac{\partial E_{\boldsymbol{X}}}{\partial \tau} = \sum_{y\in D_\tau}\frac{\partial E_{\boldsymbol{X}}}{\partial y}\cdot\frac{\partial y}{\partial \tau}$, где D_τ обозначает набор узлов, чьи выходы зависят от τ ; (7) обновление параметров τ по формуле $\Delta\tau = -\alpha\cdot\frac{\partial E_{\boldsymbol{X}}}{\partial \tau}$ в случае интерактивного обучения, то есть после предъявления каждого обучающего экземпляра или по формуле $\Delta\tau = -\alpha\cdot\frac{\partial E}{\partial \tau}$, где $E = \sum_{\boldsymbol{X}\in\{\boldsymbol{X_i}\}_{i=1}^{M}} E_{\boldsymbol{X}}$, в случае пакетного обучения, то есть после предъявления всей совокупности обучающих векторов; (8) останов алгоритма при выполнении одного из условий: $t > T$ или $E \leqslant \varepsilon$; в противном случае переход к шагу 4.

Описанный выше метод является методом градиентного спуска, в котором минимизация функционала ошибки осуществляется в пространстве настраиваемых параметров сети во время обратного прогона уточняющих сигналов. Для оптимизации стандартного метода автор системы ANFIS [17] предлагает использовать гибридное правило ее обучения, которое совмещает метод градиентного спуска и метод наименьших квадратов. С этой целью исходное множество настраиваемых параметров $a_{ij}, b_{ij}, c_{ij}, p_0^{(j)}, p_1^{(j)}\ldots, p_n^{(j)}$ декомпозируется на два подмножества, элементы одного из которых обновляются при помощи метода градиентного спуска, а элементы другого определяются при помощи метода наименьших квадратов.

5. Модель машины опорных векторов для выявления аномальных сетевых соединений. *Машина опорных векторов* является одним из широко распространенных подходов, применяемых для реше-

ния задач классификации [19], регрессии [20] и прогнозирования [21]. Метод имеет простую геометрическую аналогию, которая связана с предположением, что элементы различных классов могут быть линейно разделены как принадлежащие различным подпространствам. Множество этих элементов может быть разбито различными плоскостями, описываемыми семейством уравнений вида $\boldsymbol{W}^T \cdot \boldsymbol{X} - b = 0$ и отличающимися друг от друга вектором нормали $\boldsymbol{W}$, задающим наклон гиперплоскости, и параметром смещения b, задающим уровень подъема/спуска гиперплоскости. Пусть оптимальная гиперплоскость H_O, которая доставляет максимальное и равное расстояние между ближайшими к ней элементами из разных классов A и B, задается уравнением $\boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_O = 0$, где $\boldsymbol{W}_{\boldsymbol{O}}^T = (w_{O1}, \ldots, w_{On})^T$, а параллельные ей разделяющие гиперплоскости H_A (верхняя) и H_B (нижняя), которые проходят через эти ближайшие элементы классов A и B, задаются уравнениями $\boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_A = 0$ и $\boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_B = 0$, тогда $b_A = b_O + \varepsilon$, $b_B = b_O - \varepsilon$, где $\varepsilon > 0$. Не умаляя общности, можно считать, что $\varepsilon = 1$ (в противном случае этого можно добиться делением обеих частей уравнений на ε). Таким образом, уравнения двух гиперплоскостей H_A и H_B приобретают следующий вид: $\boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_O = 1$ и $\boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_O = -1$, а классы A и B представляются следующим образом: $A = \{\boldsymbol{X} \mid \boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_O \geqslant 1\}$, $B = \{\boldsymbol{X} \mid \boldsymbol{W}_{\boldsymbol{O}}^T \cdot \boldsymbol{X} - b_O \leqslant -1\}$.

Следовательно, модель машины опорных векторов описывается при помощи формулы:

$$Y(\boldsymbol{X}) = \operatorname{sign}\left(\sum_{i=1}^{n} w_{Oi} \cdot x_i - b_O \right).$$

Рассмотрим алгоритм обучения машины опорных векторов при условии наличия линейных гиперплоскостей H_A и H_B, корректно разделяющих все экземпляры обучающей выборки: (1) подготовка обучающих данных в виде $\left\{ (\boldsymbol{X_i}, c_i) \right\}_{i=1}^{M}$, где $c_i = [\boldsymbol{X_i} \in A] - [\boldsymbol{X_i} \in B]$; (2) нахождение множителей Лагранжа $\lambda_1^{(O)}, \ldots, \lambda_M^{(O)}$ как результат решения следующей оптимизационной задачи: $-\frac{1}{2} \cdot \sum_{i=1}^{M} \sum_{j=1}^{M} \lambda_i \cdot \lambda_j \cdot c_i \cdot c_j \cdot \boldsymbol{X_i}^T \cdot \boldsymbol{X_j} + \sum_{i=1}^{M} \lambda_i \to \max$ при ограничениях $\sum_{i=1}^{M} \lambda_i \cdot c_i = 0$ и $\lambda_i \geqslant 0$ ($i = 1, \ldots, M$); (3) вычисление вектора нормали в

уравнении гиперплоскости: $\boldsymbol{W_O} = \sum_{i=1}^{M} \lambda_i^{(O)} \cdot c_i \cdot \boldsymbol{X_i} = \sum_{j=1}^{M'} \lambda_{i_j}^{(O)} \cdot c_{i_j} \cdot \boldsymbol{X_{i_j}}$, где $\{i_1,\ldots,i_{M'}\} \subseteq \{1,\ldots,M\}$, $\left\{\boldsymbol{X_{i_j}}\right\}_{j=1}^{M'}$ — это множество опорных векторов, которым соответствуют ненулевые $\lambda_{i_j}^{(O)}$; (4) вычисление свободного коэффициента в уравнении гиперплоскости: $b_O = \boldsymbol{W_O^T} \cdot \tilde{\boldsymbol{X}} - c_i$, где $\tilde{\boldsymbol{X}}$ — один из опорных векторов; (5) уточнение модели машины опорных векторов: $Y(\boldsymbol{X}) = \operatorname{sign}\left(\sum_{j=1}^{M'} w_{i_j} \cdot \boldsymbol{X_{i_j}^T} \cdot \boldsymbol{X} - b_O\right)$, где $w_{i_j} = \lambda_{i_j}^{(O)} \cdot c_{i_j}$, и оператор суммирования берется по индексному подмножеству обучающей выборки, которое соответствует только опорным векторам $\left\{\boldsymbol{X_{i_j}}\right\}_{j=1}^{M'} \subseteq \left\{\boldsymbol{X_i}\right\}_{i=1}^{M}$.

Когда объекты из разных классов не могут быть линейно разделены, используются два подхода, причем оба из них направлены на уменьшение значения функционала эмпирического риска на элементах обучающей выборки. Первый подход заключается в применении специальных преобразований — ядер для перехода к новому пространству. Предполагается, что в новом пространстве уже будет существовать гиперплоскость, удовлетворяющая ранее заданному критерию. Второй подход основан на введении штрафной функции, чтобы игнорировать некоторые из ложно классифицируемых объектов на основе минимизации или их общего количества, или их суммарного расстояния до разделяющей гиперплоскости. В первом случае осуществляется поиск такой разделяющей гиперплоскости, которая доставляет минимальное значение следующей характеристической функции $\sum_{i=1}^{M} \left[Y(\boldsymbol{X_i}) \neq c_i\right]$. Во втором случае в роли целевой функции, также подлежащей минимизации, выступает $\sum_{i=1}^{M} \operatorname{dist}\left(\boldsymbol{X_i}, H_O\right) \cdot \left[Y(\boldsymbol{X_i}) \neq c_i\right]$, где dist(·,·) — функция расстояния между указанной парой аргументов (вектор, плоскость) в рамках заданной метрики.

6. Методика иерархической гибридизации бинарных классификаторов для выявления аномальных сетевых соединений. Общее представление предлагаемой методики для выявления аномальных сетевых соединений показано на рисунке 2 и состоит из следующих этапов: (1) построение дерева классификаторов; (2) формирование параметров сетевых соединений; (3) предобработка параметров сетевых соединений; (4) иерархический обход в ширину дерева классификаторов; (5) обнаружение и классификация сетевых аномалий.

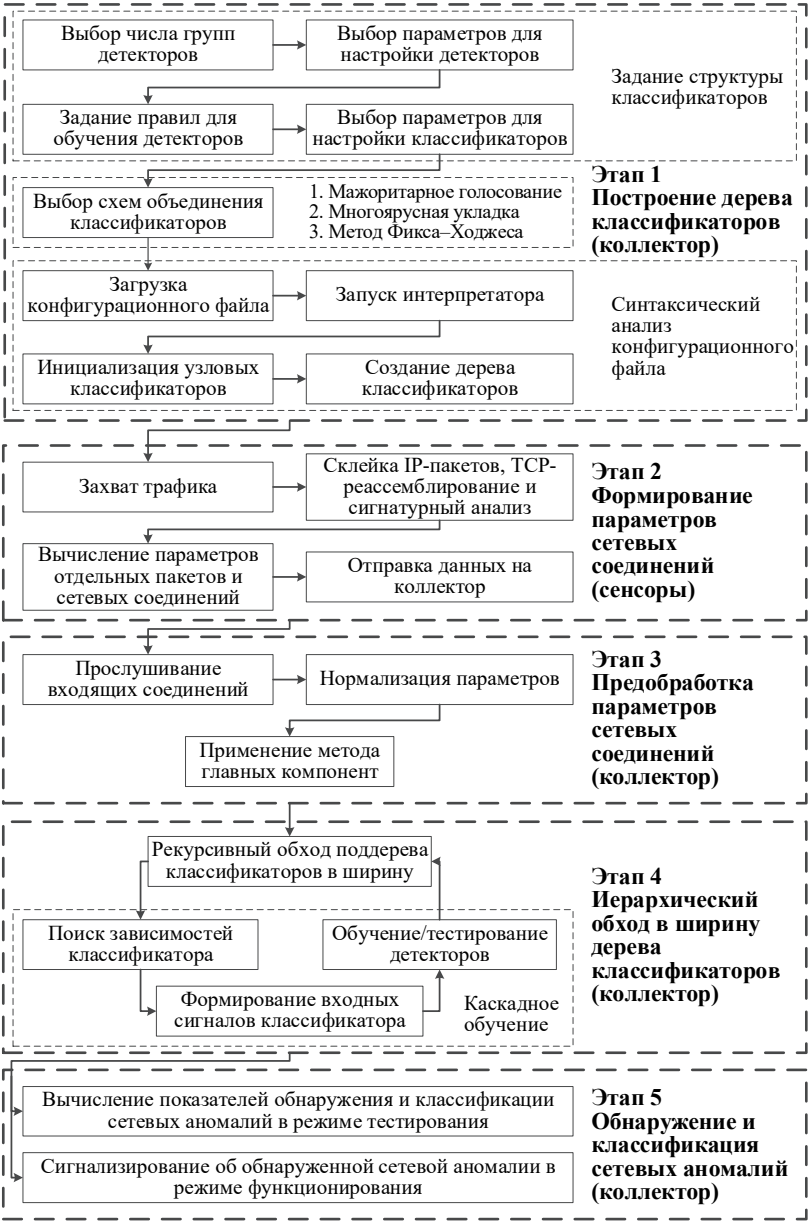

Рис. 2. Основные этапы методики иерархической гибридизации бинарных классификаторов для выявления аномальных сетевых соединений

Первый этап методики может быть охарактеризован как подготовительный, он включает в себя выбор структуры отдельных бинарных классификаторов (детекторов): размерности и числа слоев, параметров и алгоритмов обучения, типов функций активации, функций принадлежности и ядерных функций. Для каждого детектора может быть составлен набор обучающих правил. Задавая различную совокупность таких наборов правил, можно сформировать группу детекторов, каждый из которых построен на основе одной из вышеперечисленных моделей. Детекторы внутри каждой такой группы объединяются на основе подходов *один-ко-всем*, *один-к-одному* [19] или их различных производных вариаций. В первом подходе каждый детектор $F_{jk}^{(i)} : \mathbb{R}^n \to \{0,1\}\ \ (k = 1,\ldots,m)$ обучается на данных $\{(\boldsymbol{X}_l, [c_l = k])\}_{l=1}^{M}$, и функционирование группы детекторов $F_j^{(i)}$ описывается при помощи исключающего принципа:

$$F_j^{(i)}(\boldsymbol{X}) = \begin{cases} \{0\}, \text{ если } \forall k \in \{1,\ldots,m\}\ F_{jk}^{(i)}(\boldsymbol{X}) = 0 \\ \left\{k \,\middle|\, F_{jk}^{(i)}(\boldsymbol{X}) = 1\right\}_{k=1}^{m}, \text{ иначе} \end{cases}.$$

Во втором подходе каждый из $C_{m+1}^2 = \dfrac{(m+1)\cdot m}{2}$ детекторов $F_{jk_0k_1}^{(i)}$ обучается на множестве объектов, принадлежащих только двум классам с метками k_0 и k_1, — $\{(\boldsymbol{X}_l, 0) \mid c_l = k_0\}_{l=1}^{M} \bigcup \{(\boldsymbol{X}_l, 1) \mid c_l = k_1\}_{l=1}^{M}$, где $0 \leqslant k_0 < k_1 \leqslant m$, и функционирование группы детекторов $F_j^{(i)}$ может быть задано с помощью голосования max-wins:

$$F_j^{(i)}(\boldsymbol{X}) = \left\{ \arg \max_{c \in \{0,\ldots,m\}} \sum_{k=c+1}^{m} \left[F_{jck}^{(i)}(\boldsymbol{X}) = 0 \right] + \sum_{k=0}^{c-1} \left[F_{jkc}^{(i)}(\boldsymbol{X}) = 1 \right] \right\}.$$

В качестве одной из производных вариаций предыдущих подходов для комбинирования детекторов может быть упомянуто *классификационное бинарное дерево*. С экспериментальной точки зрения этот подход уже исследовался в одной из более ранних статей автора [22]. Формально такая структура задается рекурсивно следующим образом:

$$CBT_S = \begin{cases} \langle F_{jL_SR_S}^{(i)}, CBT_{L_S}, CBT_{R_S} \rangle, \text{ если } \#S \geqslant 2 \\ S, \text{ если } \#S = 1 \end{cases}.$$

Здесь $S = \{0,\ldots,m\}$ — исходный набор меток классов, $L_S \subsetneq S$ — произвольно сгенерированное или предопределенное пользователем подмножество S ($\#L_S < \#S$), $R_S = S \setminus L_S$, CBT_{L_S} — левое классификационное поддерево, CBT_{R_S} — правое классификационное поддерево, $F^{(i)}_{jL_SR_S}$ — узловой детектор, обученный на элементах множества $\{(\boldsymbol{X_l},0) \mid c_l \in L_S\}_{l=1}^{M} \bigcup \{(\boldsymbol{X_l},1) \mid c_l \in R_S\}_{l=1}^{M}$, то есть выходной результат детектора настраивается таким образом, чтобы он равнялся 0, если входной объект $\boldsymbol{X_l}$ имеет метку $c_l \in L_S$, и 1, если объект $\boldsymbol{X_l}$ имеет метку $c_l \in R_S$. Поэтому функционирование группы детекторов $F^{(i)}_j$, представленных в виде узлов такого дерева, описывается с помощью рекурсивной функции $\Phi^{(i)}_j$, задающей последовательную дихотомию множества S:

$$F^{(i)}_j(\boldsymbol{X}) = \Phi^{(i)}_j(S,\boldsymbol{X}),$$

$$\Phi^{(i)}_j(S,\boldsymbol{X}) = \begin{cases} S, \text{ если } \#S = 1 \\ \Phi^{(i)}_j(L_S,\boldsymbol{X}), \text{ если } \#S \geqslant 2 \wedge F^{(i)}_{jL_SR_S}(\boldsymbol{X}) = 0. \\ \Phi^{(i)}_j(R_S,\boldsymbol{X}), \text{ если } \#S \geqslant 2 \wedge F^{(i)}_{jL_SR_S}(\boldsymbol{X}) = 1 \end{cases}$$

Применение функции $\Phi^{(i)}_j$ к исходному набору меток классов и классифицируемому объекту позволяет осуществлять однозначный поиск метки класса этого объекта. Это объясняется тем, что поскольку по мере спуска вниз по классификационному дереву происходит дизъюнктное разбиение множества меток классов, то после достижения и срабатывания терминального детектора остается только одна возможная метка для классификации входного объекта $\boldsymbol{X}$ в качестве выходного результата $F^{(i)}_j$. Поэтому для классификационного дерева невозможны конфликтные случаи при классификации объектов, которые могут иметь место для двух других подходов комбинирования.

Другим подходом является *направленный ациклический граф*, который организует $C^2_{m+1} = \dfrac{(m+1)\cdot m}{2}$ детекторов в связную динамическую структуру, которая может быть задана следующей формулой:

$$DAG_S = \begin{cases} \langle F^{(i)}_{jSk_0k_1}, DAG_{S\setminus\{k_0\}}, DAG_{S\setminus\{k_1\}}\rangle, \text{ если } \#S \geqslant 2, \text{ где } k_0 \in S, k_1 \in S \\ S, \text{ если } \#S = 1 \end{cases}.$$

Здесь, как и в подходе один-к-одному, каждый узловой детектор $F^{(i)}_{jSk_0k_1}$ обучается на элементах $\{(\boldsymbol{X_l},0)\,|\,c_l=k_0\}_{l=1}^{M}\bigcup\{(\boldsymbol{X_l},1)\,|\,c_l=k_1\}_{l=1}^{M}$ $(k_0<k_1)$. Обход рассматриваемого графа выполняется при помощи рекурсивной функции $\Xi^{(i)}_j$, задающей поэлементное «отщепление» от множества S:

$$F^{(i)}_j(\boldsymbol{X})=\Xi^{(i)}_j(S,\boldsymbol{X}),$$

$$\Xi^{(i)}_j(S,\boldsymbol{X})=\begin{cases} S, \text{ если } \#S=1 \\ \Xi^{(i)}_j(S\setminus\{k_1\},\boldsymbol{X}), \text{ если } \#S\geqslant 2\wedge F^{(i)}_{jSk_0k_1}(\boldsymbol{X})=0. \\ \Xi^{(i)}_j(S\setminus\{k_0\},\boldsymbol{X}), \text{ если } \#S\geqslant 2\wedge F^{(i)}_{jSk_0k_1}(\boldsymbol{X})=1 \end{cases}$$

Если детектор $F^{(i)}_{jSk_0k_1}$ голосует за k_0-ый класс для объекта $\boldsymbol{X}$, то есть $F^{(i)}_{jSk_0k_1}(\boldsymbol{X})=0$, то из множества S удаляется метка k_1 как заведомо неверная, в противном случае исключается метка k_0. Процесс повторяется до тех пор, пока множество S не вырождается в одноэлементное.

В таблице 1 приведены характеристики рассмотренных схем объединения детекторов в многоклассовую модель, предназначенную для соотнесения входного объекта одной или нескольким из $(m+1)$ меток классов.

Таблица 1. Характеристики схем объединения детекторов

Схема объединения	Число детекторов, подлежащих обучению	Минимальное число детекторов, задействованных при классификации объектов	Максимальное число детекторов, задействованных при классификации объектов
Один-ко-всем	m	m	m
Один-к-одному	$\frac{(m+1)\cdot m}{2}$	$\frac{(m+1)\cdot m}{2}$	$\frac{(m+1)\cdot m}{2}$
Классификационное бинарное дерево	m	1	m
Направленный ациклический граф	$\frac{(m+1)\cdot m}{2}$	m	m

Из рассмотренных четырех схем только одна, а именно классификационное дерево, обладает переменным числом детекторов, кото-

рые могут использоваться в процессе классификации объектов. Минимальное значение достигается, когда активируется детектор $F^{(i)}_{jL_SR_S}$, расположенный в корне дерева и обученный для распознавания только одного объекта среди всех остальных, и $F^{(i)}_{jL_SR_S}(\boldsymbol{X}) = 0$ ($F^{(i)}_{jL_SR_S}(\boldsymbol{X}) = 1$), то есть когда $\#L_S = 1$ ($\#R_S = 1$). Максимальное значение достигается, когда дерево представляется последовательным списком и активируется наиболее удаленный в нем детектор. В случае сбалансированного дерева этот показатель может составлять величину $\lfloor \log_2(m+1) \rfloor$ или $\lceil \log_2(m+1) \rceil$.

На рисунке 1 представлен пример, когда каждый классификатор $F^{(i)}$ ($i = 1,\ldots,s$) содержит q_i групп $F^{(i)}_j$ ($j = 1,\ldots,q_i$), каждая из которых объединяет m детекторов $F^{(i)}_{jk}$ ($k = 1,\ldots,m$) при помощи подхода один-ко-всем. Каждая из групп детекторов $F^{(i)}_j$ обучается на различных случайных бутстреп-подвыборках, которые могут включать повторяющиеся и переупорядоченные элементы из исходного обучающего набора χ. Объединение групп $F^{(i)}_j$ в классификатор $F^{(i)}$ осуществляется на основе голосования большинством:

$$F^{(i)}(\boldsymbol{X}) = \left\{ c \left| \underbrace{\sum_{j=1}^{q_i} \left[c \in F^{(i)}_j(\boldsymbol{X}) \right]}_{\xi_i(c)} > \frac{1}{2} \cdot q_i \wedge \xi_i(c) = \max_{c' \in \{0,\ldots,m\}} \xi_i(c') \right. \right\}_{c=0}^{m}.$$

Для построения коллективного правила F [23, 24], объединяющего выходные результаты классификаторов $F^{(i)}$, были реализованы следующие подходы: *мажоритарное голосование*, представляющее собой средневзвешенное суммирование выходов отдельных классификаторов, *многоярусная укладка*, дополненная введением дополнительного атрибута — номера кластера по методу k-средних, а также *метод Фикса — Ходжеса*, представляющий собой объединение классификаторов с использованием арбитра на основе динамических областей компетентности и метода ближайших соседей. Более подробное описание этих методов с применением к задаче классификации сетевых атак представлено в [22].

Для выполнения задачи синтаксического анализа был реализован интерпретатор, поддерживающий операции условного ветвления, конкатенации векторов, векторного суммирования, покомпонентного произведения и деления. В процессе работы интерпретатора проверяется корректность обрабатываемого конфигурационного файла, и инициализируются поля объектов внутри строящегося дерева классификаторов. Данная методика подразумевает распределенную архитектуру реализующих ее систем, в которых сбор данных осуществляется вторичными узлами — сенсорами, а вся обработка агрегированных потоков данных выполняется на централизованном сервере — коллекторе.

Второй этап методики, выполняемый на стороне сенсоров, заключается в применении разработанного алгоритма сборки сырых пакетов в сетевые соединения, выделении их параметров и выполнении сигнатурного анализа с использованием нескольких разработанных параллельных модификаций алгоритмов шаблонного поиска подстроки, представленных в [25]. С этой целью было исследовано быстродействие алгоритмов Ахо — Корасик и Бойера — Мура на выбранных сигнатурных записях Snort, и реализованы их улучшенные аналоги при помощи технологий OpenMP и CUDA. Был реализован событийно-ориентированный анализатор сетевого трафика, с помощью которого было извлечено 106 сетевых параметров, среди которых можно назвать продолжительность соединения, используемую сетевую службу, интенсивность отправки хостом специальных пакетов, число активных соединений между конкретной парой IP-адресов, признак изменения масштабирования TCP-окна после фактического установления сессии, текущее состояние TCP-соединения, различные признаки наличия сканирующих пакетов на уровнях TCP, UDP, ICMP и IP и пр. Классификация этих параметров представлена на рисунке 3. Для измерения величины интенсивности отправки/приема пакетов использовался адаптированный метод скользящей средней. Суть метода заключается в разбиении заданного временного интервала $\Delta_0^{(L)} = [0, L]$ длиной L, в течение которого производится непрерывное наблюдение за рядом параметров, на несколько более мелких интервалов $\Delta_0^{(L')}, \Delta_\delta^{(L')}, \ldots, \Delta_{\delta\cdot(K-1)}^{(L')}$ одинаковой длины $0 < L' \leqslant L$, начало каждого из которых имеет смещение $0 < \delta \leqslant L'$ относительно начала предыдущего интервала. Причем $\bigcup_{i=0}^{K-1} \Delta_{\delta\cdot i}^{(L')} \subseteq \Delta_0^{(L)}$ и $\bigcup_{i=0}^{K} \Delta_{\delta\cdot i}^{(L')} \supsetneq \Delta_0^{(L)}$, поэтому $K = 1 + \left\lfloor \frac{L - L'}{\delta} \right\rfloor$. В течение промежутков времени $\Delta_0^{(L')}, \ldots, \Delta_{\delta\cdot(K-1)}^{(L')}$ де-

лаются слепки значений $\omega_0,\ldots,\omega_{K-1}$ параметров, и их средняя величина (интенсивность) ϖ в рамках временного окна длины L' рассчитывается по формуле $\varpi = \frac{1}{K}\cdot\sum_{i=0}^{K-1}\omega_i$. В данной работе использовался интервал со значением параметра L, равным пяти секундам. Длина сглаживающего интервала L' была выбрана равной одной секунде. Смещение δ было установлено в полсекунды. Предполагается, что подобный подход позволяет устранить редкие по частоте и случайные сетевые всплески и тем самым снизить число ложных срабатываний.

Рис. 3. Классификация сетевых параметров

Третий этап начинается с прослушивания входящих от сенсоров пакетов, передаваемых по протоколу RPC/SSL и содержащих вычисленные параметры соединений. Для обеспечения взаимодействия коллектора и сенсоров выбор пал в сторону именно такой связки протоколов, поскольку они гарантируют быструю и безопасную отправку данных. RPC является хорошо зарекомендовавшей себя и успешно прошедшей испытание временем технологией, которая позволяет без труда организовать компактную передачу бинарных потоков данных. А SSL, в свою очередь, широко используется для создания шифрованного канала между передатчиками данных. Перед непосредственным обучением детекторов выполняется предобработка данных параметров для уменьшения эффекта их сильной изменчивости. Многие методы, включая нейронные сети и метод главных компонент, чувствительны к

такого рода флуктуациям и требуют, чтобы все признаки обрабатываемых векторов имели одинаковый масштаб. Поэтому первый шаг предобработки каждого компонента x_{ij} вектора $\boldsymbol{X_i} \in \{\boldsymbol{X_i}\}_{i=1}^{M}$ включает его нормализацию при помощи функции $f(x_{ij}) = \dfrac{x_{ij} - x_j^{(min)}}{x_j^{(max)} - x_j^{(min)}}$ (в случае $x_j^{(max)} = x_j^{(min)}$ можно полагать $f(x_{ij}) = 0$), где $x_j^{(min)} = \min_{i=1,\ldots,M} x_{ij}$ и $x_j^{(max)} = \max_{i=1,\ldots,M} x_{ij}$. Второй шаг нормализации — уменьшение числа незначимых признаков, что достигается при помощи метода главных компонент, описываемого как последовательность следующих шагов: (1) вычисление математического ожидания случайного вектора, представленного в данном случае в виде элементов набора обучающих данных $\left\{\boldsymbol{X_i} = \left\{x_{ij}\right\}_{j=1}^{n}\right\}_{i=1}^{M}$:

$$\overline{\boldsymbol{X}} = (\overline{x}_1,\ldots,\overline{x}_n)^T = E\left[\{\boldsymbol{X_i}\}_{i=1}^{M}\right] = \frac{1}{M}\cdot\sum_{i=1}^{M}\boldsymbol{X_i} = \left(\frac{1}{M}\cdot\sum_{i=1}^{M}x_{i1},\ldots,\frac{1}{M}\cdot\sum_{i=1}^{M}x_{in}\right)^T;$$

(2) формирование элементов несмещенной теоретической ковариационной матрицы $\boldsymbol{\Sigma} = \left(\sigma_{ij}\right)_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}}$: $\sigma_{ij} = \dfrac{1}{M-1}\cdot\sum_{k=1}^{M}(x_{ki} - \overline{x}_i)\cdot(x_{kj} - \overline{x}_j)$; (3) нахождение собственных чисел $\{\lambda_i\}_{i=1}^{n}$ и собственных векторов $\{\boldsymbol{v_i}\}_{i=1}^{n}$ матрицы $\boldsymbol{\Sigma}$ как корней уравнений: $\det(\boldsymbol{\Sigma} - \lambda\cdot\mathbf{I}) = 0$ и $(\boldsymbol{\Sigma} - \lambda\cdot\mathbf{I})\cdot\boldsymbol{v} = \mathbf{0}$; (4) ранжирование собственных чисел $\{\lambda_i\}_{i=1}^{n}$ в порядке их убывания и соответствующих им собственных векторов $\{\boldsymbol{v_i}\}_{i=1}^{n}$: $\lambda_1 \geqslant \ldots \geqslant \lambda_n \geqslant 0$; (5) отбор необходимого числа $\hat{n} \leqslant n$ главных компонент: $\hat{n} = \min\left\{z \middle| g(z) \geqslant \varepsilon\right\}_{z=1}^{n}$, где $g(z) = \sum_{i=1}^{z}\lambda_i \Big/ \sum_{i=1}^{n}\lambda_i$ — мера информативности, $0 \leqslant \varepsilon \leqslant 1$ — экспертно выбираемая величина; (6) центрирование входного вектора признаков $\boldsymbol{X'}$: $\boldsymbol{X'_c} = \boldsymbol{X'} - \overline{\boldsymbol{X}}$; (7) проецирование центрированного вектора признаков $\boldsymbol{X'_c}$ в новую систему координат, задаваемую ортонормированными векторами $\{\boldsymbol{v_i}\}_{i=1}^{\hat{n}}$: $\boldsymbol{Y'} = (y'_1,\ldots,y'_{\hat{n}})^T = (\boldsymbol{v_1},\ldots,\boldsymbol{v_{\hat{n}}})^T\cdot\boldsymbol{X'_c}$, здесь $y'_i = \boldsymbol{v_i}^T\cdot\boldsymbol{X'_c}$ называется i -ой главной компонентой вектора $\boldsymbol{X'}$. Результаты экспериментов показали, что повторная нормализация после сжатия при помощи метода главных компонент необязательна.

Четвертый этап методики с точки зрения вычислительных ресурсов является наиболее трудоемким и состоит из следующих рекурсивно повторяющихся последовательностей действий: вычисление зависимостей текущего классификатора, формирование входных сигналов для текущего классификатора, обучение текущего классификатора. Была разработана специальная древовидная структура для хранения классификаторов, которая позволяет осуществлять эффективный нисходящий спуск по всем цепочкам зависимостей, начиная с верхнеуровневого классификатора до терминальных узлов, представленных детекторами. Обучение каждого классификатора порождает запрос на обучение нижележащих классификаторов, указанных в списке его зависимостей, и генерацию их выходных данных для формирования входных данных вышележащего классификатора. Следствием используемого таким образом каскадного обучения является возможность ленивой загрузки классификаторов: в обучении и распознавании участвуют только те классификаторы, которые напрямую или косвенно встречаются в списке зависимостей классификатора, ответственного за формирование общего решения в коллективе классификационных правил. Это свойство является особенно выгодным при разборе динамических правил обучения классификаторов, то есть таких правил, от успешного или неуспешного срабатывания которых зависит вызов другого правила. В частности, это характерно для классификационного дерева, когда правила являются вложенными друг в друга. Тем самым за счет применения приема ленивой загрузки удается избежать случаев бесполезного вызова того детектора, чье выходное значение, как уже известно, не повлияет на результат общего коллектива классификационных правил.

Пятый этап методики включает в себя два режима: режим оценки эффективности и режим функционирования. В первом режиме осуществляется вычисление показателей, представленных в разделе 7, во втором режиме выполняется диагностика системы без априорного знания о фактическом классе идентифицируемого сетевого соединения.

7. Результаты экспериментов. Для проведения экспериментов был использован представленный в виде pcap-файлов набор данных DARPA 1998, из которого были отобраны два класса атак типа «отказ в обслуживании», четыре класса атак, связанных со сканированием портов и хостов, и один класс, описывающий нормальные соединения. Кроме того, в данном наборе в формате csv содержатся метки классов с дополнительной метаинформацией, позволяющей сопоставить реассемблированным из сырых пакетов образам сетевых соединений при-

писанные им классы. Описание сгенерированных наборов обучающих и контрольных данных приведено в таблице 2.

Таблица 2. Обучающее и контрольное множества

Обучающее множество	
Общее количество	7000
Относительное количество уникальных записей	85.17%
Контрольное множество	
Общее количество	101113
Относительное количество уникальных записей	53.14%
Относительное количество уникальных записей, не встречавшихся в процессе обучения	47.25%

Были выбраны следующие показатели обнаружения и классификации аномальных сетевых соединений, относительно которых производилась оценка эффективности классификационных моделей: (1) уровень корректности обнаружения: $TPR = TP/(TP + FN)$, где показатель TP — число верно распознанных аномальных соединений, FN — число ошибок второго рода; (2) уровень ложных срабатываний: $FPR = FP/(FP + TN)$, где FP — число ошибок первого рода, TN — число верно распознанных нормальных соединений; (3) уровень корректности классификации: $CCR = CC_{COR}/(TP + FN + FP + TN)$, где CC_{COR} — общее число элементов, класс которых был верно определен, на объединенном наборе данных, состоящем из нормальных и аномальных соединений; (4) уровень конфликтных случаев корректной классификации: $CCR' = CC_{CONFL}/(TP + FN + FP + TN)$, где CC_{CONFL} — общее число элементов, для которых выходное значение решающего классификатора содержит несколько классов, включая верный, на объединенном наборе данных, состоящем из нормальных и аномальных соединений; (5) уровень обобщающей способности при обнаружении: $GAR = TP_{UNQ \setminus TR}/(TP_{UNQ \setminus TR} + FN_{UNQ \setminus TR})$, где показатели $TP_{UNQ \setminus TR}$, $FN_{UNQ \setminus TR}$ представляют собой число верно распознанных аномальных соединений и число ошибок второго рода соответственно, которые вычислены на уникальных данных контрольного множества, строго исключая любые данные обучающего множества; (6) уровень переобученности при обнаружении: $OVR = TPR_{UNQTR} - GAR$, где показатель TPR_{UNQTR} соответствует уровню корректности обнаружения на уникальных данных обучающего множества. Аналогично пунктам 5 и 6 могут быть введены также пока-

затели обобщающей способности и переобученности при классификации. При помощи метода главных компонент размерность пространства признаков была сокращена со 106 до 33 компонент. Полученные значения показателей эффективности приведены в таблице 3. Ячейки в первых трех колонках содержат значения соответствующих показателей эффективности, вычисленные для каждой из трех схем гибридизации. Последняя колонка соответствует среднему арифметическому (CA) показателей отдельных базовых классификаторов, представленных нейронными сетями с функциями активации типа гиперболического тангенса, нейронечеткими сетями с колоколообразными функциями принадлежности и машинами опорных векторов с радиально-базисными ядрами.

Таблица 3. Значения показателей эффективности

	Мажоритарное голосование	Многоярусная укладка	Метод Фикса — Ходжеса	CA базовых классификаторов
TPR	99.78%	99.82%	99.78%	99.3%
FPR	0.46%	2.89%	3.01%	1.32%
CCR	98.46%	97.76%	96.7%	97.9%
CCR′	0%	0%	0.01%	0.09%
GAR	99.72%	99.74%	99.72%	99.58%
OVR	0.2%	0.2%	0.2%	−0.76%

В результате применения методики гибридизации удалось повысить корректность обнаружения на 0.48% в случае объединения классификаторов при помощи мажоритарного голосования и снизить уровень ложных срабатываний на 0.86% по сравнению с усредненными показателями отдельных классификаторов. Подобный небольшой выигрыш обусловлен наличием и без того высоких показателей у отдельных классификаторов, которые представлены как группа детекторов, обученных для распознавания только одного класса атак. В случае остальных подходов для объединения классификаторов показатель ложных срабатываний существенно поднялся, сохранив при этом сравнимый с подходом мажоритарного голосования показатель корректности обнаружения. Поскольку главной целью, поставленной в данной статье, является уменьшение функционала эмпирического риска, то есть увеличение показателя CCR, то были проведены дополнительные эксперименты для более детального и точного вычисления несмещенной оценки этой характеристики. С этой целью использовалась пятиблочная кросс-валидация. Набор данных Q, содержащий 53733 неповторяющиеся записи сетевых соединений, был разбит на пять дизъюнктных подмножеств $Q_1,\ldots,Q_5$, у которых $\#Q_1 \approx \ldots \approx \#Q_5$.

Обучающая и контрольная выборки были взяты в отношении 3:2. Процесс обучения базовых классификаторов выполнялся $C_5^3 = 10$ раз при помощи множеств $\{Q_a \bigcup Q_b \bigcup Q_c\}$, где $a,b,c \in \{1,\ldots,5\} \wedge a < b < c$. В зависимости от этих множеств контрольное множество составляется следующим образом: $\{Q_d \bigcup Q_e\}$, где $d,e \in \{1,\ldots,5\} \setminus \{a,b,c\} \wedge d < e$, и на каждом из этих множеств вычисляются значения показателей $CCR_{ide}^{(\text{БК})}$ i-ого базового классификатора ($i = 1,2,3$) и $CCR_{jde}^{(\text{КП})}$ j-ого коллективного правила ($j = 1,2,3$). Итоговые значения показателей корректности классификации $CCR_i^{(\text{БК})}$ и $CCR_j^{(\text{КП})}$, которые соответствуют каждому i-ому базовому классификатору и каждому j-ому коллективному правилу, определяются следующим образом: $CCR_i^{(\text{БК})} = \frac{1}{10} \cdot \sum_{d,e \in \{1,\ldots,5\} \wedge d<e} CCR_{ide}^{(\text{БК})}$ и $CCR_j^{(\text{КП})} = \frac{1}{10} \cdot \sum_{d,e \in \{1,\ldots,5\} \wedge d<e} CCR_{jde}^{(\text{КП})}$. Аналогичным образом вычисляются величины $TPR_i^{(\text{БК})} - FPR_i^{(\text{БК})}$ и $TPR_j^{(\text{КП})} - FPR_j^{(\text{КП})}$. Полученные значения каждого из этих показателей схематически изображены на рисунке 4. Как и ранее, в этих экспериментах задействовался подход один-ко-всем на самом низком уровне для объединения детекторов в простейшую многоклассовую модель.

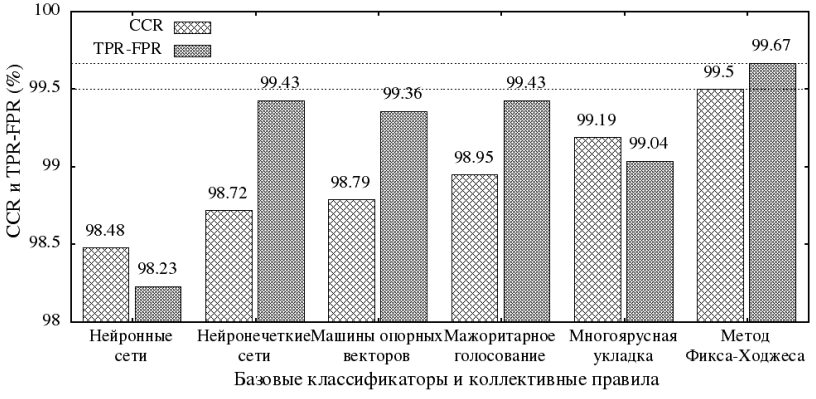

Рис. 4. Значения показателей CCR и $TPR - FPR$, полученных при помощи пятиблочной кросс-валидации

При помощи метода Фикса — Ходжеса показатель корректности классификации $CCR = 99.5\%$ был увеличен на 0.71% по сравнению с

максимальным значением этого показателя $\max_{i=1,2,3} CCR_i^{(\text{БК})} = 98.79\%$, полученного среди базовых классификаторов, а именно машинами опорных векторов. Применение метода Фикса — Ходжеса позволило повысить несущественно (на 0.24%) показатель $TPR - FPR$, представляющий собой уровень компромисса между корректностью обнаружения аномальных соединений и ложными срабатываниями.

Зависимость усредненной по десяти обучающим множествам меры информативности от числа выбранных главных компонент представлена на рисунке 5.

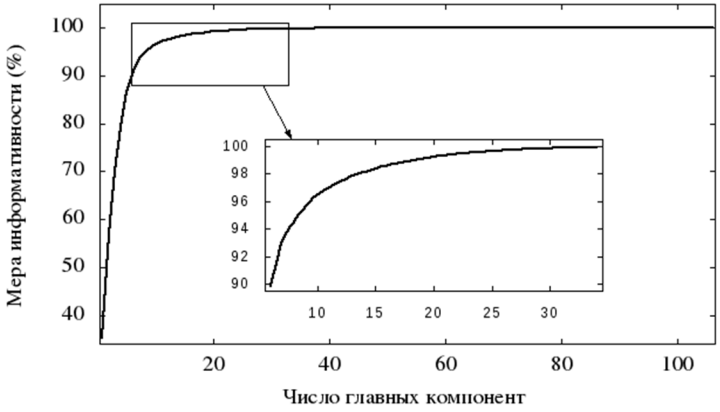

Рис. 5. Зависимость меры информативности от числа главных компонент

Из этого рисунка видно, что после получения приблизительно 30 первых главных компонент уже не наблюдается существенного прироста меры информативности, и график кривой практически полностью вырождается в постоянную функцию.

8. Заключение. В статье представлена обобщенная методика гибридизации бинарных классификаторов в рамках решения задачи выявления аномальных сетевых соединений. В качестве классификаторов, входящих в состав коллектива классификационных правил, рассмотрены нейронные сети, нейронечеткие сети и машины опорных векторов. Особенности предлагаемой методики, адаптивным ядром которой являются данные модели, — это возможность задания произвольной вложенности классификаторов друг в друга и ленивое подключение классификаторов благодаря нисходящему каскадному обучению коллектива классификационных правил. Выделены пять этапов в рассмотренной методике. На первом этапе выполняется индивидуальная настройка классификаторов и задаются правила для их обуче-

ния. Показаны примеры нескольких схем, основанных на использовании таких правил и позволяющих объединять детекторы в многоклассовую модель. Каждая из этих схем поддерживается на уровне интерпретатора разработанного программного средства, предназначенного для классификации аномальных сетевых соединений. Второй этап заключается в выполнении сигнатурного анализа содержимого отдельных и дефрагментированных пакетов, сборке сетевых соединений и извлечении из них параметров, пригодных для анализа при помощи адаптивных классификаторов. Третий этап характеризуется процессом предварительной обработки таких параметров при помощи методов нормализации и главных компонент. На четвертом этапе выполняется обход дерева классификаторов с чередованием процессов их обучения/тестирования и поиска зависимостей. Отмечено, что выполняемые в рамках этого этапа процессы являются самыми ресурсоемкими. Пятый этап — это вычисление выбранных показателей обнаружения и классификации аномальных сетевых соединений. Результаты проведенных экспериментов показали несущественный прирост показателей эффективности обнаружения и классификации сетевых аномалий, который обусловлен высоким качеством обучения базовых решателей. Несмотря на это, поставленную в исследовании цель можно считать выполненной. В дальнейших работах планируется исследовать другие наборы данных для более детального проведения экспериментов.

Литература

1. *Comer D.E.* Computer Networks and Internets: 6th edition // Pearson. 2014. 672 p.
2. *Котенко И.В., Саенко И.Б., Полубелова О.В., Чечулин А.А.* Применение технологии управления информацией и событиями безопасности для защиты информации в критически важных инфраструктурах // Труды СПИИРАН. 2012. Вып. 1(20). С. 27–56.
3. *Браницкий А.А., Котенко И.В.* Построение нейросетевой и иммуноклеточной системы обнаружения вторжений // Проблемы информационной безопасности. Компьютерные системы. 2015. № 4. С. 23–27.
4. *Branitskiy A., Kotenko I.* Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers // The 18th IEEE International Conference on Computational Science and Engineering (IEEE CSE2015). 2015. 152–159.
5. *Amini M., Rezaeenour J., Hadavandi E.* Effective Intrusion Detection with a Neural Network Ensemble using Fuzzy Clustering and Stacking Combination Method // Journal of Computing and Security. 2015. vol. 1. no. 4. pp. 293–305.
6. *Wang G., Hao J., Ma J., Huang L.* A New Approach to Intrusion Detection using Artificial Neural Networks and Fuzzy Clustering // Expert Systems with Applications. 2010. vol. 37. no. 9. pp. 6225–6232.
7. *Chandrasekhar A.M., Raghuveer K.* Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM Classifiers // International Conference on Computer Communication and Informatics (ICCCI). 2013. pp. 1–7.
8. *Saied A., Overill R.E., Radzik T.* Detection of Known and Unknown DDoS Attacks using Artificial Neural Networks // Neurocomputing. 2016. vol. 172. pp. 385–393.

9. *Agarwal B., Mittal N.* Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques // Procedia Technology. 2012. vol. 6. pp. 996–1003.
10. *He H.T., Luo X.N., Liu B.L.* Detecting Anomalous Network Traffic with Combined Fuzzy-Based Approaches // International Conference on Intelligent Computing. 2005. pp. 433–442.
11. *Колмогоров А.Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Докл. АН СССР. 1957. Т. 114. № 5. С. 953–956.
12. *Cybenko G.* Approximation by Superpositions of a Sigmoidal Function // Mathematics of control, signals and systems. 1989. vol. 2. no. 4. pp. 303–314.
13. *Hornik K., Stinchcombe M., White H.* Multilayer Feedforward Networks are Universal Approximators // Neural networks. 1989. vol. 2. no. 5. pp. 359–366.
14. *Funahashi K.I.* On the Approximate Realization of Continuous Mappings by Neural Networks // Neural networks. 1989. vol. 2. no. 3. pp. 183–192.
15. *Riedmiller M., Braun H.* A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm // Proceedings of IEEE International Conference On Neural Networks. 1993. pp. 586–591.
16. *Fahlman S.E.* Faster-learning variations on Back-propagation: An empirical study // Proceedings of the 1988 Connectionist Models Summer School. 1988. pp. 38–51.
17. *Jang J.-S.R.* ANFIS: Adaptive-Network-Based Fuzzy Inference System // IEEE Transactions on Systems, Man, and Cybernetics. 1993. vol. 23. no. 3. pp. 665–685.
18. *Takagi T., Sugeno M.* Fuzzy Identification of Systems and Its Applications to Modeling and Control // IEEE Transactions on Systems, Man, and Cybernetics. 1985. vol. SMC-15. no. 1. pp. 116–132.
19. *Hsu C.W., Lin C.J.* A Comparison of Methods for Multiclass Support Vector Machines // IEEE transactions on Neural Networks. 2002. vol. 13. no. 2. pp. 415–425.
20. *Drucker H. et al.* Support Vector Regression Machines // Advances in Neural Information Processing Systems. 1997. vol. 9. pp. 155–161.
21. *Müller K.R. et al.* Predicting Time Series with Support Vector Machines // Proceedings of International Conference on Artificial Neural Networks. 1997. pp. 999–1004.
22. *Branitskiy A., Kotenko I.* Hybridization of Computational Intelligence Methods for Attack Detection in Computer Networks // Journal of Computational Science. 2016. (В печати).
23. *Zhou Z.H.* Ensemble Methods: Foundations and Algorithms // CRC press. 2012. 218 p.
24. *Городецкий В.И., Серебряков С.В.* Методы и алгоритмы коллективного распознавания: обзор // Труды СПИИРАН. 2006. Т. 1. №. 3. С. 139–171.
25. *Браницкий А.А.* Архитектура распределенной системы обнаружения, классификации и предотвращения сетевых атак на основе сигнатурного анализа и методов вычислительного интеллекта // Материалы 9-й конференции «Информационные технологии в управлении» (ИТУ-2016). СПб.: ОАО «Концерн «ЦНИИ «Электроприбор». 2016. С. 651–655.

Браницкий Александр Александрович — младший научный сотрудник лаборатории проблем компьютерной безопасности, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, искусственный интеллект, функциональное программирование. Число научных публикаций — 17. branitskiy@comsec.spb.ru, http://www.comsec.spb.ru; 14-я линия В.О., 39, Санкт-Петербург, 199178; р.т.: +7(812)328–7181, Факс: +7(812)328–4450.

Поддержка исследований. Работа выполнена при финансовой поддержке гранта РНФ 15-11-30029.

DOI 10.15622/sp.52.10

A.A. BRANITSKIY

HIERARCHICAL HYBRIDIZATION OF BINARY CLASSIFIERS FOR DETECTING ANOMALOUS NETWORK CONNECTIONS

Branitskiy A.A. **Hierarchical Hybridization of Binary Classifiers for Detecting Anomalous Network Connections.**

Abstract. The paper considers a generalized hybrid approach for constructing a set of classification rules through the example of detection of anomalous network connections. There are five stages in the proposed technique. The first stage involves the setting of adaptive classifiers. At the second stage the signature analysis, creation of network connections and formation of network parameters are performed. The third stage is preprocessing of network parameters. At the fourth stage bypassing of a classifier tree in width is performed together with training or testing. The fifth stage is a detection of anomalous network connections. The distinctive features of the proposed technique are the possibility to set an arbitrary nesting of classifiers in each other and a lazy involvement of classifiers due to descending cascade learning of a general classifier fusion. The results of the experiments with the use of an open data set for calculating the performance rates of detection and classification of network anomalies are provided.

Keywords: network anomalies, network connections, TCP/IP protocols, classifier hybridization.

Branitskiy Alexander Alexanderovich — junior researcher of computer security problems laboratory, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS). Research interests: security of computer networks, artificial intelligence, functional programming. The number of publications — 17. branitskiy@comsec.spb.ru, http://www.comsec.spb.ru; 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone: +7(812)328–7181, Fax: +7(812)328–4450.

Acknowledgements. This research is supported by RSF (grant 15-11-30029).

References

1. Comer D.E. Computer Networks and Internets: 6th edition. Pearson. 2014. 672 p.
2. Kotenko I.V., Saenko I.B., Polubelova O.V., Chechulin A.A. [The use of information and security event management technology to protect the information in critical infrastructures]. *Trudy SPIIRAN – SPIIRAS Proceedings.* 2012. vol. 1(20). pp. 27–56. (In Russ.).
3. Branitskiy A.A., Kotenko I.V. [Construction of neural network and immune cell intrusion detection system]. *Problemy informacionnoj bezopasnosti. Komp'yuternye sistemy – Problems of information security. Computer systems.* 2015. vol. 4. pp. 23–27. (In Russ.).
4. Branitskiy A., Kotenko I. Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers. The 18th IEEE International Conference on Computational Science and Engineering (IEEE CSE2015). pp. 152–159.
5. Amini M., Rezaeenour J., Hadavandi E. Effective Intrusion Detection with a Neural Network Ensemble using Fuzzy Clustering and Stacking Combination Method. *Journal of Computing and Security*. 2015. vol. 1. no. 4. pp. 293–305.
6. Wang G., Hao J., Ma J., Huang L. A New Approach to Intrusion Detection using Artificial Neural Networks and Fuzzy Clustering. *Expert Systems with Applications*. 2010. vol. 37. no. 9. pp. 6225–6232.

7. Chandrasekhar A.M., Raghuveer K. Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM Classifiers. International Conference on Computer Communication and Informatics (ICCCI). 2013. pp. 1–7.
8. Saied A., Overill R.E., Radzik T. Detection of Known and Unknown DDoS Attacks using Artificial Neural Networks. *Neurocomputing*. 2016. vol. 172. pp. 385–393.
9. Agarwal B., Mittal N. Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques. *Procedia Technology*. 2012. vol. 6. pp. 996–1003.
10. He H.T., Luo X.N., Liu B.L. Detecting Anomalous Network Traffic with Combined Fuzzy-Based Approaches. International Conference on Intelligent Computing. 2005. pp. 433–442.
11. Kolmogorov A.N. [On the representation of continuous functions of several variables as superpositions of continuous functions of one variable and addition]. *Dokl. AN SSSR – Proceedings of the USSR Academy of Sciences.* 1957. vol. 114. no. 5. pp. 953–956. (In Russ.).
12. Cybenko G. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of control, signals and systems*. 1989. vol. 2. no. 4. pp. 303–314.
13. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators. *Neural networks*. 1989. vol. 2. no. 5. pp. 359–366.
14. Funahashi K.I. On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural networks*. 1989. vol. 2. no. 3. pp. 183–192.
15. Riedmiller M., Braun H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. Proceedings of IEEE International Conference On Neural Networks. 1993. pp. 586–591.
16. Fahlman S.E. Faster-learning variations on Back-propagation: An empirical study. Proceedings of the 1988 Connectionist Models Summer School. 1988. pp. 38–51.
17. Jang J.-S.R. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*. 1993. vol. 23. no. 3. pp. 665–685.
18. Takagi T., Sugeno M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetics*. 1985. vol. SMC-15. no. 1. pp. 116–132.
19. Hsu C.W., Lin C.J. A Comparison of Methods for Multiclass Support Vector Machines. *IEEE transactions on Neural Networks*. 2002. vol. 13. no. 2. pp. 415–425.
20. Drucker H. et al. Support Vector Regression Machines. *Advances in Neural Information Processing Systems*. 1997. vol. 9. pp. 155–161.
21. Müller K.R. et al. Predicting Time Series with Support Vector Machines. Proceedings of International Conference on Artificial Neural Networks. 1997. pp. 999–1004.
22. Branitskiy A., Kotenko I. Hybridization of Computational Intelligence Methods for Attack Detection in Computer Networks. *Journal of Computational Science*. 2016. (In print).
23. Zhou Z.H. Ensemble Methods: Foundations and Algorithms. CRC press. 2012. 218 p.
24. Gorodetsky V.I., Serebryakov S.V. [Methods and Algorithms of the Collective Recognition: A Survey]. *Trudy SPIIRAN – SPIIRAS Proceedings.* Issue 2006. 1. vol. 3. pp. 139–171. (In Russ.).
25. Branitskiy A.A. [Architecture of distributed system of detection, classification and prevention of network attacks based on signature analysis and computational intelligence methods]. *Materialy 9-j konferencii «Informacionnye tekhnologii v upravlenii» (ITU-2016)* [Information technologies in management (ITU-2016)]. *SPb.: OAO "Koncern "CNII "Ehlektropribor"*. 2016. pp. 651–655. (In Russ.).