ПРОГРАММИРОВАНИЕ, 2022, № 6, с. 3–13

АНАЛИЗ ДАННЫХ

УДК 004.421.6

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ ОТБОРА ПРИЗНАКОВ СЕТЕВОГО ТРАФИКА ПРИ ОБНАРУЖЕНИИ КОМПЬЮТЕРНЫХ АТАК

© 2022 г. В. В. Беликов[a,*] (ORCID: 0000-0003-1423-1072)

[a] МИРЭА – Российский технологический университет
119333 Москва, проспект Вернадского, д. 78, Россия
**E-mail: belikov_v@mirea.ru*

Поступила в редакцию 28.04.2022 г.
После доработки 21.06.2022 г.
Принята к публикации 07.07.2022 г.

В статье предложено решение задачи отбора признаков сетевого трафика с использованием методов глубокого обучения с подкреплением, представляющее классификацию в виде последовательного процесса, на каждом шаге которого принимается решение о достаточности наличия имеющихся значений признаков для соотнесения объекта с классом. Предложенное решение позволяет варьировать количество используемых признаков от одного экземпляра к другому. Проведенный эксперимент продемонстрировал возможность использования такого решения для увеличения обобщающей способности моделей классификации и снижения переобучения при их использовании в СОВ сетевого типа для обнаружения компьютерных атак, в том числе при наличии только несбалансированных обучающих наборов данных.

DOI: 10.31857/S0132347422060024

1. ВВЕДЕНИЕ

Компьютерные системы оказывают все большее влияние на современную жизнь, что делает кибербезопасность важной областью исследований. Среди различных инструментов обеспечения защиты компьютерных сетей как одного из основных компонентов компьютерных систем ключевую роль играют системы обнаружения вторжений (intrusion detection systems, СОВ). Недостаточная эффективность применения сигнатурного анализа, в том числе его ограниченные возможности при обнаружении неизвестных ранее компьютерных атак, а также бурное развитие методов интеллектуального анализа данных являются причинами большого числа исследований, посвященных использованию альтернативного подхода, закладываемого в основе СОВ: подхода, основанного на методах машинного обучения. Вместе с тем, на успешное применение разработанных с использованием методов машинного обучения решений накладывает ограничения особенности предметной области обнаружения компьютерных атак в сети: отсутствие или ограниченный объем имеющихся наборов реальных данных; несбалансированная обучающая выборка; высокая вариативность сетевого трафика и постоянное совершенствование способов проведения компьютерных атак.

2. АНАЛИЗ И ПРОБЛЕМА

Проблема отбора признаков сетевого трафика для обнаружения компьютерных атак. Задачу обнаружения компьютерных атак в контексте применения методов машинного обучения чаще всего представляют как задачу бинарной классификации объекта, извлеченного из имеющихся данных. СОВ сетевого типа используют в качестве источника данных сетевой трафик, в основе которого лежат пакеты. Пакеты являются основными единицами сетевого взаимодействия и представляют из себя оформленные блоки данных, состоящие из служебной информации (например, флаг TCP/UDP) и полезной нагрузки (payload), формат которой определяется используемым протоколом передачи данных. Чаще всего при этом объектом классификации выступает поток, который является набором пакетов, отражающим сетевую среду в течение определённого интервала времени. Однако бинарный формат и большой объем пакетов в рамках одного потока делает невозможным их применение в методах машинного обучения напрямую, что обуславливает необходимость предварительного конструирования вектора признаков, которые отражают либо статистические свойства потока (например, доля флагов TCP, средняя длина полезной нагрузки), либо свойства потока как последовательности пакетов;

наиболее распространенным является первый подход, исследованию которого и посвящена данная статья. Существует несколько практических инструментов, позволяющих решать указанную задачу, например, Argus[1], CICFlowMeter[2], NFStream[3], Fullstats[4]. Общее количество признаков, извлекаемых указанными инструментами, может достигать 85. Однако использование всего набора признаков на этапе эксплуатации модели приводит к задержке реакции СОВ.

Кроме этого, достаточно хорошо освещенной проблемой является недостаточное качество имеющихся обучающих выборок, выражающееся в наличии следующих недостатков:

• ввиду отсутствия реальных наборов данных, не выкладываемых из соображений приватности, использование обучающих выборок приводит к переобучению модели, к ее слабой адаптации для применения в отношении сетевого трафика, по своим характеристикам отличающегося от используемого для обучения модели;

• несбалансированность выборки, подавляющая часть которой является обычным трафиком, усложняет обучение, характеризующееся большим количеством ошибок второго рода.

Высокая вариативность сетевого трафика и постоянное совершенствование способов проведения компьютерных атак исключают возможность по созданию универсальной обучающей выборки, устраняющей вышеперечисленные недостатки, и обосновывают необходимость увеличения обобщающей способности моделей классификации при их использовании в СОВ сетевого типа. Одним из подходов для решения указанной проблемы является отбор признаков сетевого трафика.

Математическая постановка задачи. Задача классификации сетевого трафика для обнаружения компьютерных атак с использованием СОВ может быть формализована следующим образом. Пусть задано множество объектов сетевого трафика $\mathcal{X}$, представленных вектором из d признаков $\mathbf{x} = (x_1,\ldots,x_d)$, множество классов $\mathcal{Y} = \{0$ – безопасный сетевой график, 1 – компьютерная атака$\}$ и множество моделей (гипотез) $\mathcal{H}$, описываемых в виде функции, которая каждому объекту ставит в соответствие один из классов $h : \mathcal{X} \to \mathcal{Y}$. Тогда истинная ошибка модели $h \in \mathcal{H}$ определяется как неотрицательная функция потерь $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$. Для бинарной классификации функция потерь задается равной единице, если гипотеза неправильно определила класс, и нулю в ином случае:

$$\ell(h(\mathbf{x}), y) = \begin{cases} 1, & \text{если } h(\mathbf{x}) \neq y \\ 0, & \text{иначе} \end{cases} \tag{2.1}$$

При заданном вероятностном распределении $\mathcal{D}$ над $\mathcal{X} \times \mathcal{Y}$ функция ℓ является случайной, а ее математическое ожидание именуется истинной ошибкой (true risk) модели h [1]:

$$L_D(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(h(\mathbf{x}), y)]$$

Так как в большинстве случаев распределение $\mathcal{D}$ неизвестно, то модель выбирается по имеющейся выборке из m объектов $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}^m, y^m))$ на основе минимизации эмпирической ошибки (empirical risk), рассчитываемой как среднее арифметическое наблюдаемых значений функции потерь ℓ:

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}\ell(h(\mathbf{x}^i), y^i) \tag{2.2}$$

Вместе с тем, использование вместо истинной ошибки эмпирической в качестве целевого показателя при решении задачи классификации ввиду ограниченного размера обучающей выборки часто приводит к переобучению. Для численной оценки переобучения может быть использовано взятое в отношении случайной выборки S математическое ожидание ошибки обобщения, определяемое как разница между истинной и эмпирической ошибкой модели h [2]:

$$L_{\mathcal{D}}(h) - L_S(h) \tag{2.3}$$

Различным семействам моделей H соответствуют различные значения $\sup_{h\in H}\left|L_{\mathcal{D}}(h) - L_S(h)\right|$, определяемыми характеристикой этого семейства, например, VC-размерность [3]. Одним из наиболее известных и теоретически изученных подходов, используемых для уменьшения энтропии, является минимизация структурного риска (structural risk minimization), при котором для набора вложенных семейств моделей $H_1 \subset H_2 \subset \ldots \subset H_r \subset$ ищется компромисс между сложностью модели и эмпирической ошибкой за счет добавления регуляризационной функции $F(h)$, описывающей сложность семейства, к которому эта модель принадлежит:

$$\arg\min_h L_S(h) + F(h) \tag{2.4}$$

Набор семейств $\{H_r\}$, $H_1 \subset H_2 \subset \ldots$, при имеющейся возрастающей последовательности положительных чисел $a_1 < a_2 < \ldots$ может быть получен заданием H_r как множества моделей, среднее количество используемых признаков которых меньше или равно a_r, а в качестве характеристики сложности H_r, используемой в регуляризационной функции $F(h)$ – значение a_r.

[1] https://openargus.org/
[2] https://github.com/ahlashkari/CICFlowMeter
[3] https://www.nfstream.org
[4] https://www.cl.cam.ac.uk/research/srg/netos/projects/brasil/

Для использования в таком подходе модель должна быть расширена до набора двух функций:

$$h = (h_y, h_z), \quad h_y : \mathcal{X} \to \mathcal{Y}, \quad h_z : \mathcal{X} \to \mathcal{Z} \qquad (2.5)$$

Здесь функция h_y ставит в соответствие объекту $\mathbf{x}$ предполагаемый класс y, а функция h_z – вектор $\mathbf{z} \in \mathcal{Z} = \{0,1\}^m$, где i-й элемент вектора $z_i = 1$, если i-й признак использовался для предсказания класса y. Тогда выражение (2.4) принимает вид:

$$\begin{aligned} &\arg\min_h L_S(h_y) + \lambda \frac{1}{m}\sum_{i=1}^{m} \|h_z(\mathbf{x}^i)\|_0 = \\ &= \arg\min_h \frac{1}{m}\sum_{i=1}^{m} [\ell(h_y(\mathbf{x}^i), y^i) + \lambda \|h_z(\mathbf{x}^i)\|_0] \end{aligned} \qquad (2.6)$$

Варьирование значения λ позволяет выбирать компромисс между ограничением среднего количества используемых признаков и эмпирической ошибкой классификации. Таким образом, это позволяет с использованием отбора наиболее значимых признаков повышать обобщающую способность модели без увеличения размера обучающей выборки m в системах обнаружения вторжений.

Еще одной проблемой описываемой предметной области является несбалансированность классов, которая заключается в том, что в имеющейся выборке, как правило, количество объектов сетевого трафика, соответствующих компьютерным атакам, значительно меньше количества объектов безопасного сетевого трафика $|\{(\mathbf{x}^i, y^i) \in S : y^i = 1\}| \ll$ $\ll |\{(\mathbf{x}^i, y^i) \in S : y^i = 0\}|$. Модели, построенные на основе такой выборки с использованием алгоритма, предназначенного для обучения на сбалансированном наборе, характеризуются критически маленьким значением полноты, так как с большей вероятностью относят новые наблюдения к классам, представленным большим числом обучающих примеров. Одним из наиболее известных подходов, позволяющим обойти описанную проблему без изменения пропорций классов в имеющейся выборке является классификация с использованием издержек [4]. При таком подходе применяется иная функция потерь, которая каждому виду ошибки классификации: первого и второго рода – ставит в соответствие стоимости: $C_{10} \in \mathbb{R}_+$ и $C_{01} \in \mathbb{R}_+$.

$$\ell(h_y(\mathbf{x}), y) = \begin{cases} C_{01}, & \text{если } h_y(\mathbf{x}) = 0 \text{ и } y = 1 \\ C_{10}, & \text{если } h_y(\mathbf{x}) = 1 \text{ и } y = 0 \\ 0, & \text{иначе} \end{cases} \qquad (2.7)$$

Функция потерь вида (2.7) является расширением функции потерь вида (2.1) и может быть сведена к ней при выполнении условия $C_{01} = C_{10} = 1$. При известных значениях C_{01} и C_{10}, определяемых предметной областью, cost-sensitive learning может использоваться для решения задач с несбалансированным набором данных. Если эти значения неизвестны, то они могут быть заданы пропорционально объему экземпляров каждого из классов в имеющейся выборке S:

$$\begin{aligned} C_{01} &= \frac{|\{(\mathbf{x}^i, y^i) \in S : y^i = 0\}|}{m} \\ C_{10} &= \frac{|\{(\mathbf{x}^i, y^i) \in S : y^i = 1\}|}{m} \end{aligned} \qquad (2.8)$$

Таким образом, поставленной в статье задачей является нахождение модели, которая является решением оптимизационной задачи (2.6) в отношении функции потерь вида (2.7)

3. ОБЗОР РАБОТ ПО ТЕМАТИКЕ ПРОЕКТА

В работе [5] представлен обзор существующих методов отбора признаков сетевого трафика для систем обнаружения вторжений; осуществлено их обобщение в три категории: фильтры (filter), основанные на показателях, не зависящих от метода классификации; методы обертки (wrapper), где значимость признаков основывается на результатах применения методов классификации для их разных комбинаций; гибридные методы. Проведено сравнение эффективности и производительности этих методов с использованием набора данных KDD 1999, для оценки использовались показатели полнота (Recall)

$$\text{Recall} = \frac{TP}{TP + FN}$$

и доля ложноположительных результатов (FPR)

$$FPR = \frac{FP}{FP + TN}$$

Здесь TP – количество истинно положительных результатов, TN – количество истинно отрицательных результатов, FP – количество ложноположительных результатов, FN – количество ложноотрицательных результатов. Сделан вывод об обоснованности дальнейшего развития методов отбора признаков сетевого трафика для систем обнаружения вторжений.

В работе [6] рассмотрено применение искусственных нейронных сетей в системах обнаружения вторжений; в основу отбора признаков сетевого трафика было положено использование искусственной нейронный сети с добавленным зашумленным узлом, где для оценки значимости признака использовалось значение показателя отношения сигнал-шум (signal-to-noise ratio, SNR):

$$SNR_i = 10\log_{10}\left(\frac{\sum_{j=1}^{J}(w_{itp,j}^1)^2}{\sum_{j=1}^{J}(w_{N,j}^1)^2}\right)$$

Таблица 1. Параметры первого правила Snort для обнаружения атаки DNS spoofing

Имя параметра правила	Значение параметра правила
flow	to_client
content	|85 80 00 01 00 01 00 00 00 00|
content	|C0 0C 00 0C 00 01 00 00 00|<|00 0F|
fast_pattern	only

Здесь SNR_i – значение отношения сигнал–шум для i-го признака, J – количество узлов в скрытых слоях, $w^1_{i,j}$ – значения веса связи первого слоя между узлом i и узлом j, $w^1_{N,j}$ – значения веса связи первого слоя между зашумленным узлом N и узлом j. Экспериментальная оценка проведена с использованием сбалансированной выборки, сформированной на основе набора данных CDX.

В работе [7] статистические признаки сетевого трафика были обогащены биграммами, извлеченными из полезной нагрузки. Для отбора признаков сетевого трафика разработан метод рекурсивного добавления признаков (RFA) при обучении SVM – такой метод отличается от методов обертки тем, что отбор признаков происходит непосредственно на этапе обучения классификатора. Проведен эксперимент с использованием набора данных ISCX 2012.

В работе [8] предложен метод отбора признаков CFS-BA, представляющий из себя комбинацию основанного на корреляции фильтра (CFS) и метаэвристического алгоритма глобальной оптимизации Bat. Проведен эксперимент на наборах данных NSL-KDD, AWID, and CICIDS2017, который позволил авторам сделать вывод о повышении значений метрик-curacy и F-меры без повышения значения FPR при применении предложенного метода отбора признаков, где:

$$F_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{TP}{TP + FN}$$

Наконец в работе [9] для формирования признакового пространства оценка значимости и отбор признаков набора данных CICIDS2017 использовался энтропийный подход с последующим применением корреляционного анализа, что позволило количество признаков сократить с 84 до 10. Для устранения дисбаланса классов применялся метод занижения доли объектов определенного класса случайным сэмплированием (undersampling). Было осуществлено сравнение различных методов классификации в части обнаружения web-атак, по результатам которого был выбран RandomForest. Полученные результаты апробации обученной с использованием RandomForest модели на сетевом траффике, собранным авторами в реальной сети, продемонстрировали крайне низкие значения F-меры, несмотря на высокие значения показателей классификации для тестовой выборки из набора CICIDS2017. На основании полученных результатов авторами был сделан вывод о влиянии отличий в физической структуре сетей и настройках оборудования на возникновение ошибок классификатора и точность модели.

Однако в исследованиях, проводимых на данную тему, целью являлся отбор признаков для всего набора данных, что может быть рассмотрено как задача нахождения условного экстремума функции (2.2) с учетом ограничения на количество используемых признаков. Вместе с тем, ввиду как большого различия между разными видами сетевых атак, так и значимого разнообразия реализаций одного конкретного вида, для обнаружения вторжения с достаточным уровнем значения показателя эффективности классификации для каждого отдельного экземпляра сетевого трафика количество используемых признаков может значительно варьироваться. Так, для системы обнаружения вторжений Snort[5] в отношении одного и того же вида атаки – DNS spoofing – существует два правила обнаружения, каждое из которых является достаточным для сигнализирования о соответствующей атаке. При этом для выполнения первого достаточно найти в любом месте полезной нагрузки одного из пакетов входящего сетевого потока два шаблона выражения (табл. 1), тогда как для второго правила требуется нахождения трех шаблонов выражений, на расположение которых внутри полезной нагрузки накладываются ограничения вроде смещения относительно начала нагрузки (offset), допустимого расстояния между шаблонами (distance) и соответствия значения отдельных байтовых полей заданным тестам(byte_test) (табл. 2).

Решение оптимизационной задачи, приведенной в выражении (2.6), сокращает именно среднее, а не максимальное количество признаков сетевого трафика, что позволяет при необходимости ограничиваться небольшим количеством признаков для простых объектов классификации и использовать большее количество признаков для сложных.

Также в проведенных исследованиях проблема дисбаланса классов решалась с использованием методов занижения или завышения (oversampling) доли объектов определенного класса. В отличие от этого, использование функции ошибки, приведенной в выражении (2.7), позволяет решать напрямую проблему дисбаланса классов без изменения состава обучающей выборки.

[5] https://www.snort.org/

В работе [10] было впервые предложено использовать для отбора признаков целевую функцию вида (2.6). Для нахождения ее решения использовался алгоритм Q-learning с линейной аппроксимацией, что сделало применение предложенного метода ограниченным.

В работе [11] указанная целевая функция использовалась для решения задачи классификации с признаками, добывание которых требует затрат (classification with costly features); линейная аппроксимация в методе Q-learning была заменена полносвязной нейронной сетью. Однако в указанной работе не рассматривался вопрос о связи сокращения среднего количества используемых признаков с переобучением. Кроме этого, для эксперимента использовались сбалансированные наборы данных и для его оценки применялся показатель *Accuracy*, что делает проблематичными перенос полученных результатов на несбалансированные наборы данных.

4. МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ

Нахождение решения, представленного в виде (2.5), требует представления процесса классификации объекта сетевого трафика в виде последовательного принятия решения, где на каждом шаге необходимо выбрать либо запрос дополнительного, ранее неизвестного модели значения признака, либо соотнесение с классом. Тогда такой процесс может быть описан как взаимодействие классификатора, выступающего в роли агента, с эпизодическим марковским процессом принятия решений, который задается набором следующих элементов:

• Множество состояний (наблюдений состояния) среды $s \in \mathcal{S}$, описываемых конкатенацией двух векторов:

$$s = (\mathbf{x} \| \mathbf{z}) \tag{4.1}$$

Здесь вектор $\mathbf{x} \in \mathcal{R}^d$ представляет из себя значения известных признаков; вектор $\mathbf{z} \in \{0,1\}^d$ определяет, какие признаки известны классификатору ($z_i = 1$, если i-й признак известен классификатору, $z_i = 0$ иначе).

• Множество доступных действий агента $a \in \mathcal{A}$, которое включает в себя множество действий на добывание значения признака (выбор признака, значение которого будет запрашиваться у среды) и множество действий классификации.

• Распределение вероятностей перехода на шаге t в состояние s' и получения награждения r при выполнении действия a в состоянии s, заданных на множестве $\mathcal{S} \times \mathcal{A}$:

$$p(s', r | s, a) = P[S^t = s', R^t = r | S^{t-1} = s, A^t = a]$$

Таблица 2. Параметры второго, альтернативного правила Snort для обнаружения атаки DNS spoofing

Имя параметра правила	Значение параметра правила
flow	to_client
content	|81 80|
depth	4
offset	2
byte_test	2,>,0,0,relative,big
byte_test	2,>,0,2,relative,big
content	|00 00 00 00|
within	–4
distance	4
content	|C0 0C 00 01 00 01|
distance	0
byte_test	4,<,61,0,relative,big
byte_test	4,>,0,0,relative,big

При получении действия на добывание значения i-го признака среда возвращает состояние $s^{t+1} = (\mathbf{x} \| \mathbf{z})$, отличающееся от предыдущего состояния наличием значения x_i i-го признака, а также изменением значения z_i с 0 на 1. Возвращаемое вознаграждение $r_t = -\lambda$ при этом является фактически штрафом за использование дополнительного признака, величина которого определяется параметром λ из выражения (2.6).

При получении действия классификации y_{pred} среда переходит в терминальное состояние, обозначающее конец эпизода, а возвращаемое вознаграждение определяется в зависимости от истинной категории y_{true}:

$$r_t = -l(y_{\text{pred}}, y_{\text{true}}) \tag{4.2}$$

Действия агента выбираются в соответствии со стохастической стратегией, задаваемой условным распределением, определяющим вероятность действия a при условии нахождения в состоянии s:

$$\pi(a|s) = P(A^t = a | S^{t-1} = s)$$

Так как пространство состояний может быть большим или бесконечным, то вместо табличного метода представления используется представление в виде параметризованной функции $\pi : \Theta \times \times \mathcal{S} \times \mathcal{A} \to [0,1]$, задающей семейство функций $\{\pi_\theta\}_{\theta \in \Theta}$, таких что $\sum_{a \in \mathcal{A}} \pi_\theta(a|s) = 1$ для каждого s. Как правило, для параметризации стратегий агента используются искусственные нейронные сети.

Реализацией эпизода управляемого марковского процесса является траектория

$$\tau = (s_0, a_1, s_1, r_1, a_2, \ldots, s_{T-1}, a_T, s_T, r_T)$$
$$a_t \sim \pi(\cdot|s_{t-1}), (s_t, r_t) \sim p(\cdot|s_{t-1}, a_t) \quad (4.3)$$

Стратегия агента π вместе с вероятностями перехода марковского процесса принятия решения задают вероятностное распределение на множестве траекторий. Траектория численно характеризуется суммой полученных наград $R(\tau) = \sum_{k=0}^{T} r_k$, являющейся случайной величиной.

Для оценки стратегии агента используется оценочная функция состояния (value function) $V : \mathcal{S} \to \mathbb{R}$, рассчитываемая как математическое ожидание суммы наград по всем возможным траекториям, начинающимся с состояния s:

$$V^{\pi}(s) = \mathop{\mathbb{E}}_{\tau \sim p,\pi}[R(\tau)|S_0 = s] \quad (4.4)$$

Одним из наиболее известных результатов динамического программирования является доказательство существования и единственности при определенных условиях оптимальной оценочной функции $V^*(s) = \max_{\pi} V_{\pi}(s)$ для всех s. В работе [10] показано, что при задании марковского процесса принятия решения указанным выше образом решением уравнения (2.6) является оптимальная стратегия агента π^*, соответствующая оптимальной оценочной функции $V^* = V^{\pi^*}$. Доказательство остается верным и при замене функции потерь вида (2.1), используемой в указанной работе, на функцию потерь вида (2.7).

В случае большого размера пространства состояний $\mathcal{S}$, а также когда неизвестны вероятности перехода $p(s', r|s, a)$, для нахождения оптимальной стратегии агента π^* используются методы обучения с подкреплением, для которых достаточно наличия среды или ее имитационной модели, в ответ на текущее состояние s и выбранное агентом действия a сэмплирующей новое состояние s' и награду r.

В работе [11] для нахождения оптимальной стратегии использовался алгоритм Deep Q-learning. Однако в основе этого алгоритма заложена стохастическая аппроксимация как метод нахождения решения уравнения оптимальности Беллмана, позволяющая на каждой итерации распространять обновление только на один шаг назад. Это ограничение делает метод Deep Q-learning непрактичным в отношении марковского процесса принятия решений с “разреженной” наградой, когда наибольшее награждение выдается в конце эпизода, что справедливо для описываемой предметной области, где сумма вознаграждений за эпизод $R(\tau)$ в большей степени зависит от корректности классификации, являющейся финальным действием эпизода. Существующая модификация Retrace(lambda) [12], направленная на обход указанного ограничения, позволяет распространить обновление только до тех пор, пока действия используемой для исследования стратегии совпадают с оптимальными действиями обучаемой “жадной” стратегии.

В текущей работе использовался PPO с обрезанной суррогатной целевой функцией (proximal policy optimization with clipped surrogate objective), являющийся современным представителем группы методов, которые напрямую оптимизируют стратегию (policy-based) [13]. Данная группа методов является методами online policy, в которых отсутствует различение между обучаемой стратегией и стратегией, используемой для исследования, что позволяет выбирать требуемую глубину обновления для каждой итерации и, как следствие, обходить проблему “разреженной награды”. Выбор именно алгоритма PPO обосновывался стабильностью его работы; малым числом гиперпараметров; несложностью в реализации при одновременном обеспечении уровня эффективности, соответствующего другим современным методам обучения с подкреплением.

При использовании PPO на каждой итерации k обновления параметризованной стратегии π_{θ_k} после выполненных заданного количества шагов взаимодействия со средой используется несколько итераций метода градиентного спуска для нахождения решения оптимизационной задачи со следующей суррогатной целевой функцией, рассчитываемой в применении к полученному набору $\mathcal{D}_k = \{\tau\}$ траекторий вида (4.3):

$$L^{CLIP}(\theta_k, \theta) = \frac{1}{|\mathcal{D}_k T|} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} L^{CLIP}_{\tau,t}(\theta) \quad (4.5)$$

$$L^{CLIP}_{\tau,t}(\theta) = \min(r_t(\theta)\hat{A}_t^{\pi_{\theta_k}}, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$$
$$r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \quad (4.6)$$

В формуле (4.6) функция $\mathrm{clip}(x, x_{\min}, x_{\max})$ ограничивает элемент x отрезком допустимых значений $[x_{\min}, x_{\max}]$, задавая таким образом доверительную область, а ϵ выступает в роли гиперпараметра. Для расчета статистической оценки значения Advantage функции $A_t^{\pi} = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$ для t-го состояния имеющейся траектории используется формула обобщенной оценки (Generalized Advantage Estimation – GAE), позволяющая с использованием гиперпараметров γ и μ варьировать и находить баланс между значениями смещения и дисперсии (bias/variance tradeoff).

$$\hat{A}_t^{(\gamma,\mu)} = \sum_{;=0}^{\infty} (\gamma\mu)^l \delta^V_{t+1}, \quad (4.7)$$

где

$$\delta_t^V = -V^{\pi_{\theta_k}}(s_t) + r_t + \gamma V^{\pi_{\theta_k}}(s_{t+1})$$

В свою очередь для расчета значения функции ценности $V^{\pi_{\theta_k}}$ используется также аппроксимация на основе нейронной сети V^θ (critic – "критик"), часто первые слои которой разделяют архитектуру и веса нейронной сети, используемой для аппроксимации стратегии агента. Обновление весов нейронной сети на каждой итерации k достигается минимизацией следующей функции

$$L^V(\theta) = \frac{1}{|\mathcal{D}_k T|}\sum_{\tau\in\mathcal{D}}\sum_{t=0}^{T}\left(V^\theta(s_t) - \sum_{i=t}^{T} r_i\right)^2 \qquad (4.8)$$

Выбор необходимой степени исследования новых для агента действий (exploration/explotation tradeoff) возможен максимизацией функции энтропии агента, который рассчитывается как статистическая оценка средней по всем посещаемым состояниям отрицательной энтропии стратегии агента

$$L^{ENT}(\theta) = \frac{1}{|\mathcal{D}_k T|}\sum_{\tau\in\mathcal{D}}\sum_{t=0}^{T}\sum_{a\in\mathcal{A}}\pi^\theta(a|s)\log\pi^\theta(a|s) \qquad (4.9)$$

Так как градиентный спуск является наиболее ресурсоемкой операцией, то в одной из самых популярных версий алгоритма PPO, представленной ниже, оптимизация выражений (4.5), (4.8), (4.9) происходит одновременно за счет их суммирования в единую целевую функцию с использованием соответствующих коэффициентов k^V и k^{ENT}, также выступающих в роли гиперпараметров.

Алгоритм 1. Алгоритм PPO с ограничивающей суррогатной целевой функцией

Входные данные: θ_0 – начальные параметры стратегии

Цикл $i = 0,1,2,\ldots$ **выполнять**

Получить для стратегии $\pi_k = \pi_{\theta_k}$ набор траекторий $\mathcal{D}_k = \{\tau\}$

Рассчитать значения $\hat{A}_t^{\pi_k}$ с использованием формулы (4.7)

Выполнить обновление параметров стратегии θ_k с использованием K шагов метода стохастического градиентного спуска

$$L(\theta) = L^{CLIP}(\theta_k,\theta) - k^V L^V(\theta) - k^{ENT} L^{ENT}(\theta)$$

$$\theta_{k+1} \leftarrow \arg\max_\theta L$$

Конец цикла

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНОГО АНАЛИЗА

Проведение эксперимента включало в себя два этапа:

• обучение с использованием алгоритма 1 модели и ее тестирование на наборе данных CIC-IDS2017 [14];

• апробация обученной модели в отношении реального траффика, используемого в работе [9].

При проведении эксперимента для обеспечения возможности апробации обученной модели в отношении реального траффика использовался набор данных CICIDS2017, в части нелегитимного сетевого траффика включающий только атаки грубого перебора (Brute Force), межсайтовый скриптинг (XSS) и Sql-иньекции, представляющие из себя наиболее известные примеры реализации уязвимостей веб-приложений [15]. Предобработка данных проводилась в порядке, указанном в [9], за исключением того, что дополнительно осуществлялись: выбор пересечения множеств признаков из набора CICIDS2017 и набора траффика, полученного на реальной сетевой инфраструктуре; исключение признаков с нулевой дисперсией ввиду их полной неинформативности; Z-нормализация оставшихся данных. После случайного перемешивания предобработанный набор был разделен на три выборки: обучающую (4096 объектов), валидационную (990 объектов) и тестовую (2181 объект). Общее количество признаков: 38.

Для проведения эксперимента была создана программная реализация формализованной выше среды марковского процесса принятия решения, совместимая с OpenAI gym. Исходный код среды и программного обеспечения, используемого для проведения эксперимента, доступны в репозитории

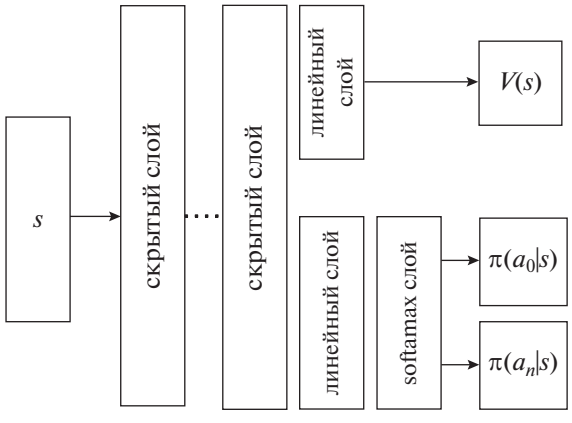

Рис. 1. Архитектура используемой нейронной сети.

Таблица 3. Найденные оптимальные значения гиперпараметров

Название гиперпараметра	Найденное оптимальное значение
Гиперпараметры градиентного спуска PPO	
Размер батча данных	1024
Коэффициент скорости обучения	0.0001 с линейным убыванием
Пороговое значение коэффициента нормирования градиента	1.0
Количество эпох	20
Гиперпараметры алгоритма PPO	
Количество шагов взаимодействия с каждой отдельной средой	128
Количество параллельно запущенных сред	16
Порог обрезания ε	0.6
Коэффициент k^V оптимизируемой функции "критика" L^V	0.8
Коэффициент k^{ENT} оптимизируемой функции энтропии L^{ENT}	0.075
Параметр длины горизонта μ	0.6
Гиперпараметры нейронной сети	
Количество скрытых слоев	3
Размер скрытого слоя	256

`https://github.com/james116blue`

Значения коэффициентов $C_{01} = 0.7$ и $C_{10} = 0.3$ выражения (2.7), определяющих в среде марковского процесса принятия решения награду за действие, соответствующее неправильной классификации, рассчитывались по формуле (2.8). Обучение производилось для следующих значений λ целевой функции (2.6), определяющих абсолютную величину отрицательного вознаграждения, выдаваемого средой в МППР при осуществлении агентом действия на добывания признака $r_i = -\lambda$: 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001. Для каждого значения λ обучалось пять моделей, соответствующих разным случайным инициализациям (random seed), с использованием валидационной выборки отбиралась лучшая. Сбор траекторий для ускорения процесса обучения осуществлялся на 16 параллельных средах.

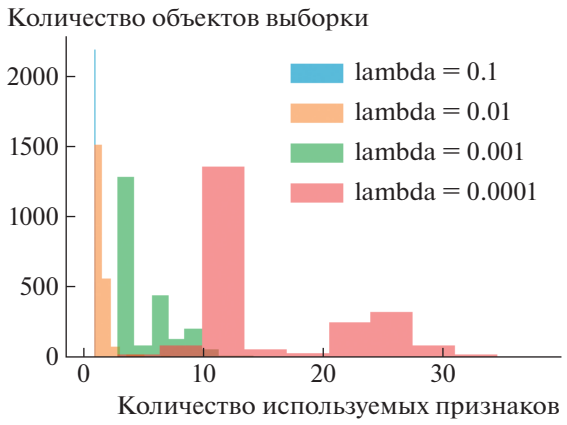

Рис. 2. Гистограмма количества используемых для классификации признаков.

Архитектура нейронной сети, используемой для аппроксимации стратегии и функции ценности, включала в себя несколько скрытых слоев, на первый из которых поступал вектор, описывающий состояние агента s (рис. 1). Так как на каждом шаге эпизода набор доступных действий зависел от того, значения каких признаков уже известны, то дополнительно перед слоем softmax использовалось маскирование недоступных действий $\mathbf{o} = \mathbf{u} - 10^6(0,0\|\mathbf{z})$, где $\mathbf{u}$ – выходные значения линейного слоя, находящегося перед softmax слоем, $\mathbf{o}$ – входные значения softmax слоя, $\mathbf{z}$ – вектор известных признаков состояния агента. Два нуля (0, 0), конкатенируемые с вектором $\mathbf{z}$, использовались для указания агенту о возможности выбора действий классификации. Коэффициент 10^6 давал для действий, соответствующим уже известным признакам ($z_i = 1$), нулевую вероятность $\pi(a_i|s) = e^{o_i}\left(\sum_j e^{o_j}\right)^{-1}$, обусловленную точностью чисел с плавающей точкой в программном пакете pytorch. Количество скрытых слоев и их размерность являлись гиперпараметрами модели. На каждом линейном слое в качестве функции активации использовалась ReLU $f(x) = \max(0, x)$.

Для выбора оптимальных значений гиперпараметров использовался применительно к обучающей выборке метод Tree-structured Parzen Estimator Approach (TPE) [16] с ранним окончанием обучения на основе медианных оценок значений

Таблица 4. Результаты эксперимента

Среднее количество используемых признаков	Точность	Полнота	F-мера
1	0.806	0.915	0.857
1	0.86	0.894	0.877
1.41	0.953	0.87	0.909
1.42	0.956	0.882	0.918
4.87	0.953	0.876	0.913
7.05	0.961	0.882	0.92
15.37	0.949	0.883	0.915

модели на валидационной выборке для коэффициента $\lambda = 0.01$ целевой функции (2.6). Полученные значения гиперпараметров представлены в таблице 3.

Полученные результаты оценки модели на тестируемой выборке набора данных CICIDS2017 представлены в табл. 4.

Как видно из полученных результатов, изменение значения λ позволяет варьировать между высоким значением эффективности классификации объектов и малым количеством используемых для этого признаков. При этом для любого значения λ имеет место следующее: для экземпляров, принадлежность к классу определить которых определяется более комплексной зависимостью, модель может запрашивать большее количество признаков. В отношении других экземпляров модели достаточно использования меньшего количества признаков. Это также продемонстрировано на гистограмме количества используемых признаков для каждого классифицируемого объекта – рис. 2. Например, для $\lambda = 0.0001$ максимальное количество используемых признаков может доходить до 40, тогда как модой является значение 10.

При этом для разных значений λ множество наиболее часто используемых признаков и их порядок могут отличаться, что демонстрируется на рис. 3, 4.

Так как в работе [9] сравнивались только модели малослойного (shallow) обучения, то на обучающей выборке набора данных CICIDS2017 также было осуществлено обучение глубокой полносвязной нейронной сети, в которой были реализованы следующие механизмы, направленные на минимизацию переобучения и повышения обобщающей способности модели:

- раннее прерывание обучения с использованием валидационной выборки [17];
- L2 регуляризация весов нейронной сети [18];
- случайное исключение отдельных нейронов (dropout) [19].

Параметры указанных механизмов, а также количество и размерность скрытых слоев и коэффициент скорости обучения являлись гиперпараметрами, значения которых подбирались на валидационной выборке. В итоге оценка модели на тестовой выборке CICIDS2017 указала значение показателя F-мера равным 0.94.

Результаты апробации модели, полученной с использованием алгоритма 1 на траффике реальной сетевой инфраструктуры, показали, что уменьшение среднего количества используемых

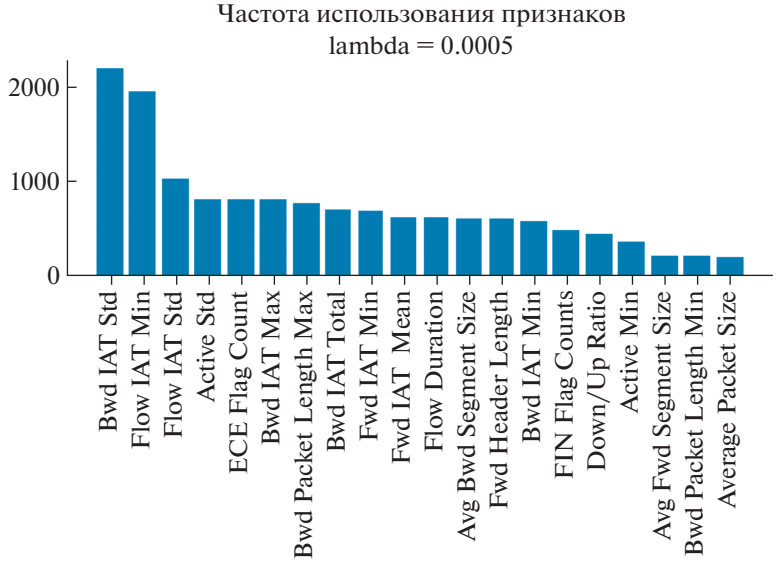

Рис. 3. Частота использования признаков для $\lambda = 0.0005$.

Рис. 4. Частота использования признаков для $\lambda = 0.0001$.

для классификации признаков приводит к снижению значения показателя F-меры, рассчитанной на выборке, полученной тем же порядком, что и обучающая (из набора данных CICIDS2017), но одновременно позволяет значительно повысить значение показателя F-меры, рассчитанной на выборке, по своим характеристикам отличающейся от обучающей. Об этом наглядно свидетельствует сравнение полученных значений F-меры с результатами применения метода Random Forest ($F_1 = 0.043$), описанных в статье [9], а также сравнение с результатом апробации глубокой полносвязной нейронной сети, показавшей себя немногим лучше модели, обученной с использованием алгоритма Random Forest ($F_1 = 0.075$) – рис. 5.

Таким образом, с использованием отбора только тех признаков, которые требуются для достижения компромисса между ограничением среднего количества используемых признаков и эмпирической ошибкой классификации, заданного в выражении 6, можно достичь повышения обобщающей способности модели и соответственно снижения эффекта переобучения. Это может быть объяснено в том числе тем, что повышение значения показателя F-меры достигается повышением точности за счет уменьшения полноты как одного из сопутствующих эффектов переобучения – рис. 6.

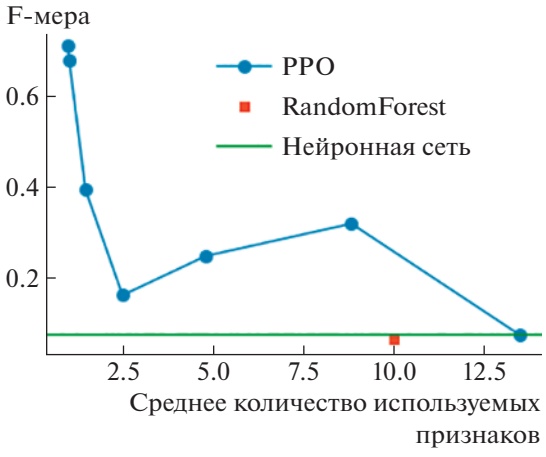

Рис. 5. Результаты апробирования моделей на выборке, полученной на реальной сетевой инфраструктуре.

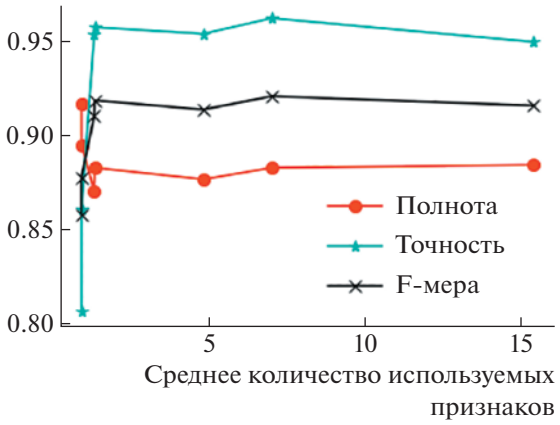

Рис. 6. Зависимость показателей классификации от среднего количества используемых признаков, оцененные по тестовой выборке набора данных CICIDS2017.

6. ВЫВОДЫ И ПРОДОЛЖЕНИЕ РАБОТЫ

Таким образом, в статье предложено решение задачи отбора признаков сетевого трафика с использованием методов глубокого обучения с подкреплением, представляющее классификацию в виде последовательного процесса, на каждом шаге которого принимается решение о достаточности наличия имеющихся значений признаков для соотнесения объекта с классом. Указанное решение позволяет варьировать количество используемых признаков от одного экземпляра к другому. Проведенный эксперимент продемонстрировал возможность использования такого решения для увеличения обобщающей способности моделей классификации и снижении переобучения при их использовании в СОВ сетевого типа для обнаружения компьютерных атак, в том числе при наличии только несбалансированных обучающих наборов данных.

СПИСОК ЛИТЕРАТУРЫ

1. *Shalev-Shwartz S., Ben-David S.* Understanding machine learning: From theory to algorithms. Cambridge university press. 2014. 445 p.
2. *Hardt M., Recht B., Singer Y.* Train faster, generalize better: Stability of stochastic gradient descent // International Conference on Machine Learning. 2016. P. 1225–1234.
3. *Vapnik V., Levin E., Cun Y.L.* Measuring the VC-Dimension of a Learning Machine // Neural Computation. 1994. V. 6. № 5. P. 851–8761.
4. *Ling C.X., Sheng V.S.* Cost-sensitive learning and the class imbalance problem // Encyclopedia of machine learning. 2011. P. 231–235.
5. *Lipmaa H., Yung M., Lin D.* Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection Syste // International Conference on Information Security and Cryptology. 2006. P. 153–167.
6. *Moore K.L., Bihl T.J., Bauer K.W.* Feature extraction and feature selection for classifying cyber traffic threats // The Journal of Defense Modeling and Simulation. 2017. V. 14. № 3. P. 217–231.
7. *Hamed T., Dara R., Kremer S.C.* Network intrusion detection system based on recursive feature addition and bigram technique // Computers & security. 2018. V. 73. P. 137–155.
8. *Zhou Y., Cheng G., Jiang S., Dai M.* Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier // Computer networks. 2020. V. 174. P. 107–123.
9. *Горюнов М.Н., Мацкевич А.Г., Рыболовлев Д.А.* Синтез модели машинного обучения для обнаружения компьютерных атак на основе набора данных CIC-IDS2017 // Труды Института системного программирования РАН. 2020. Т. 32. № 5. С. 81–94.
10. *Dulac-Arnold G., Denoyer L., Preux P., Gallinari P.* Datum-wise classification: a sequential approach to sparsity // InJoint European conference on machine learning and knowledge discovery in databases. 2011. P. 375–390.
11. *Janisch J., Pevny T., Lisy V.* Classification with costly features using deep reinforcement learning // InProceedings of the AAAI Conference on Artificial Intelligence. 2019. V. 33. P. 3959–3966.
12. *Hernandez-Garcia J.F., Sutton R.S.* Understanding multi-step deep reinforcement learning: a systematic study of the DQN target. arXiv preprint. arXiv:1901.07510. 2019.
13. *Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O.* Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. 2017.
14. Intrusion Detection Evaluation Dataset (CIC-IDS2017). https://www.unb.ca/cic/datasets/ids- 2017.htm. 2017.
15. *Лесько С.А.* Модели и сценарии реализации угроз для интернет-ресурсов // Russian Technological Journal. 2020. Т. 8. № 6. С. 9–33.
16. *Bergstra J., Bardenet R., Bengio Y., Kegl B.* Algorithms for hyper-parameter optimization. // Advances in neural information processing systems. 2011. V. 24. P. 123–145.
17. *Prechelt L.* Early stopping-but when? // InNeural Networks: Tricks of the trade. 1998. P. 55–69.
18. *Krogh A., Hertz J.* A simple weight decay can improve generalization // Advances in neural information processing systems. 1991. V. 4. P. 230–245.
19. *Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.* Dropout: a simple way to prevent neural networks from overfitting // The journal of machine learning research. 2014. V. 15. № 1. P. 29–58.