DOI: 10.18137/RNU.V9187.25.03.P.106 УДК 004.942

**Легкодумов Александр Алексеевич**
аспирант, МИРЭА – Российский технологический университет, Москва. ORCID: 0000-0002-2562-4333
Электронный адрес: studkkso0416@mail.ru
**Alexander A. Legkodumov**
Postgraduate, MIREA – Russian Technological University, Moscow. ORCID: 0000-0002-2562-4333
E-mail address: studkkso0416@mail.ru

**Козеев Борис Николаевич**
главный специалист АО «Альфа-Банк»; аспирант, МИРЭА – Российский технологический университет, Москва. ORCID: 0009-0009-0993-8082
Электронный адрес: kozeev.boris2018@yandex.ru
**Boris N. Kozeev**
Chief specialist of JSC "Alpha-Bank"; Postgraduate, MIREA – Russian Technological University, Moscow. ORCID: 0009-0009-0993-8082
E-mail address: kozeev.boris2018@yandex.ru

**Беликов Владимир Вячеславович**
кандидат военных наук, доцент базовой кафедры № 252, МИРЭА – Российский технологический университет, Москва. ORCID: 0000-0003-1423-1072
Электронный адрес: belikov_v@mirea.ru
**Vladimir V. Belykov**
Ph.D. of Military Sciences, Associate Professor of the Basic Department No. 252, MIREA – Russian Technological University, Moscow. ORCID: 0000-0003-1423-1072
E-mail address: belikov_v@mirea.ru

**Корольков Андрей Вячеславович**
кандидат технических наук, заведующий базовой кафедрой № 252, МИРЭА – Российский технологический университет, Москва. SPIN-код: 3849-6868, AuthorID: 551555
Электронный адрес: korolkov@mirea.ru
**Andrey V. Korolkov**
Ph.D. of Engineering Sciences, Head of the Basic Department No. 252, MIREA – Russian Technological University, Moscow. SPIN-code: 3849-6868, AuthorID: 551555
E-mail address: korolkov@mirea.ru

# ОБЗОР ТРЕНИРОВОЧНЫХ ОКРУЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

**Аннотация.** В статье рассматриваются основные составляющие тренировочных окружений для обучения автономных агентов. Данные агенты используются для обучения одного из направлений искусственного интеллекта – обучения с подкреплением, основы которого также изложены в статье. Автономные агенты применяются для реагирования на инциденты безопасности в инфраструктуре, что нивелирует возможные потери. Агент следует оптимальной политике, которую получает после обучения в тренировочном окружении в виде определенного аспекта информационной безопасности. Создание данных окружений является многосоставным процессом, поэтому перед созданием собственного окружения необходимо выделить ключевые значимые компоненты имеющихся тренировочных окружений. Цель статьи – выделение ключевых критериев тренировочных окружений ACD, аспектов их работы, соответствующей приближенности окружения к реальности, для достоверности

и валидности полученных политик агентов в ситуациях противодействия злоумышленнику в реальных информационных системах. Научная новизна работы заключается в комплексной систематизации существующих подходов к исследованию тренировочных окружений, выявлении компонентов тренировочных окружений и нюансов их функционирования. В статье определены основы обучения с подкреплением и указаны основы, протекающие в процессе обучения автономных агентов. Рассмотрено новое явление в традиционной информационной защите – автоматизированная информационная защита (Automated Cyber Defense), частью которой являются тренировочные окружения и автономные агенты. Приведены достоинства и недостатки симуляторов и эмуляторов. Продемонстрировано, что для определенной задачи в информационной безопасности необходимо использовать определенное тренировочное окружение. Приведено введение в обучение с подкреплением и дана формальная постановка задачи в обучении с подкреплением и в исследуемой области. Получены основные составляющие тренировочных окружений, которые могут быть применены для дальнейшего создания собственного тренировочного окружения.

***Ключевые слова:*** обучение с подкреплением, тренировочное окружение, автономный агент, автоматизированная информационная защита.

***Для цитирования:*** *Легкодумов А.А., Козеев Б.Н., Беликов В.В., Корольков А.В.* Обзор тренировочных окружений с использованием обучения с подкреплением // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2025. № 3. С. 106 – 124. DOI: 10.18137/RNU.V9187.25.03.P.106

## OVERVIEW OF ACO GYM USING REINFORCEMENT LEARNING

**Abstract.** The article discusses the main components of ACO Gyms for training autonomous agents. These agents are used to train one of the directions of artificial intelligence – reinforcement learning, the basics of which are also outlined in the article. Autonomous agents are used to respond to security incidents in the infrastructure, thereby leveling out potential losses. The agent follows an optimal policy, which it receives after training in ACO Gyms in the form of a certain aspect of information security. Creating these environments is a multicomponent process, so before creating your own environment it is necessary to highlight the key significant components of existing training environments. The purpose of the article is to highlight the key aspects of ACD training environments, aspects of their work, corresponding proximity of the environment to reality, for reliability and validity of the obtained policies of agents in situations of counteraction to an attacker in real information systems. The scientific novelty of the work consists in a comprehensive systematization of existing approaches to the research of ACD training environments, identification of components of training environments and nuances of their operation. The paper defines the foundations of reinforcement learning and specifies the fundamentals proceeding in the process of training autonomous agents. A new phenomenon in traditional information defense is considered – Automated Cyber Defense, part of which are ACD training environments and autonomous agents. The advantages and disadvantages of simulators and emulators are presented. It is demonstrated that a specific training environment should be used for a specific task in information security. An introduction to reinforcement learning is given and a formal problem formulation in reinforcement learning and the domain under study is given. The main components of ACD training environments are derived, which can be applied to further create one's own training environment.

***Keywords:*** reinforcement learning, ACO Gyms, autonomous agent, Automated Cyber Defense.

***For citation:*** Legkodumov A.A., Kozeev B.N., Belikov V.V., Korolkov A.V. (2025) Overview of ACO Gym using reinforcement learning. *Vestnik of Russian New University. Series: Complex Systems: Models, analysis, management*. No. 3. Pp. 106 – 124. DOI: 10.18137/RNU.V9187.25.03.P.106 (In Russian).

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

### *Введение*

В современном быстроменяющемся мире проблема информационной безопасности приобретает важное значение. С 2020 года вопросы информационной безопасности стоят особенно остро, а число атак злоумышленников увеличивается. Они могут быть направлены на разные объекты – от некритической инфраструктуры и персональных компьютеров до критической информационной инфраструктуры и больничных аппаратов. Их целью выступают не только персональные компьютеры пользователей, но и крупные сети частных и государственных учреждений. Количество таких атак увеличивается каждый год, что отражается на возможностях специалистов информационной безопасности. Эти причины создают потребность в стандартизированных инструментах оценки тренировочных сред для подготовки ИБ-специалистов. Существуют различные варианты ACD [1], так как эта область включает в себя многие другие области.

Методы подобных атак постоянно совершенствуются, превосходя способы защиты от них, поскольку злоумышленники используют атаки с применением средств автоматизации, в том числе с использованием искусственного интеллекта (далее – ИИ) [1].

В настоящее время существуют решения, использующие машинное обучение для определения вредоносного трафика в сети [2]. Такой выбор области информационной безопасности для имплементации классического машинного обучения является эффективным.

Применение ИИ в информационной безопасности является важным шагом в развитии средств защиты. Однако эффективность автономных агентов напрямую зависит от качества тренировочных окружений (ACTO), для которых до сих пор отсутствуют формальные критерии оценки. В ближайшем будущем на основе ИИ появятся:

- программно-аппаратные комплексы;
- антивирусы;
- IPS/IDS-системы [3];
- межсетевые экраны и др.

Для уменьшения количества или предотвращения атак используются аналогичные решения безопасности – автоматизированная информационная защита, которая:

- повышает уровень защищенности системы;
- совершенствует защиту с учетом возникающих новых угроз.

*Автоматизированная информационная защита* (Automated Cyber Defense – ACD) является новым этапом развития классической информационной защиты, базирующейся на эвристике и собранных ранее паттернах злоумышленников. ACD – это система принятия решений с возможностями, приближенными к экспертному уровню, принцип работы которой приближен к человеческому обучению [1]. Автоматизированная информационная защита включает автоматизированные решения безопасности, в том числе решения в области работы с ИИ, для создания автоматизированных защитных агентов, которые формируют политику противодействия атакам злоумышленников. Настоящее исследование систематизирует ключевые требования к ACTO, включая:

- наличие эмулятора/симулятора;
- обязательность сценариев;
- уровни реалистичности;
- гибкость целей агентов;
- стандартизированную конфигурацию.

Агент учится принимать решения с помощью обучения с подкреплением – одного из методов машинного обучения, не требующего заранее подготовленных и размеченных данных, на собственном опыте, следовательно, необходимо верно определить действия, ограничения и установить точную логику коммуникации агента со средой [4]. Указанные действия требуют наличия строгой оценки аспектов тренировочного окружения для успешного применения полученной политики в реальной инфраструктуре.

Использование ИИ в информационной безопасности является одним из аспектов автоматизированной информационной защиты. На текущий момент существуют решения, основанные на машинном обучении, позволяющие определить трафик злоумышленника в сети [2], однако в ситуации атаки они не всегда являются подходящими.

Такое соединение информационной безопасности и обучения с подкреплением является на текущий момент очень популярным направлением [5]. Исследователи изучают эффективность данного подхода, возможность его применения в различных аспектах информационной безопасности [6; 7] с помощью проведения соревнований по созданию наиболее эффективного синего агента и апробацией полученных результатов в реальных системах.

### *Методы*

Теоретико-методологической основой исследования выступают исследования зарубежных и отечественных исследователей, связанные с обучением с подкреплением, и их применение в информационной безопасности. В исследовании использовались общенаучные методы системного анализа и синтеза, контент-анализа.

### *Обучение с подкреплением*

Обучение с подкреплением может применяться для решения задач из различных аспектов информационной безопасности [8], например, при решении следующих задач:

- наличие в пакетах трафика SQL-инъекций [9];
- анализ трафика [10];
- тестирование проникновений [11].

Обучение с подкреплением (Reinforcement Learning) – направление в машинном обучении, в котором осуществляется динамическая настройка параметров стратегии выбора действий, выполняемых агентом [12].

В каждый момент времени агент находится в определенном состоянии, действует, переходя в другое состояние, и получает награду – вещественное число. Задача агента – максимизировать награду за некоторый длительный промежуток времени.

Формально обучение с подкреплением можно определить как марковский процесс принятия решений (MDP), задаваемый кортежем $(S, A, P, R, \gamma)$, где

- $S$ – пространство состояний;
- $A$ – пространство действий;
- $P(s' \mid s, a)$ – функция переходов;
- $R(s, a, s')$ – функция вознаграждения;
- $\gamma \in [0;1]$ – коэффициент дисконтирования [12].

Оптимальная политика $\pi : S \to A$ максимизирует ожидаемую дисконтированную награду: $V^{\pi}(s) = \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right]$.

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

В работе Е.В. Орловой проведено исследование обучения с подкреплением в сфере социально-экономических задач и отмечено, что «в большинстве современных приложений RL обучение производится на имитированном опыте, поэтому можно сгенерировать неограниченное количество данных с меньшими затратами, чем получение реальной информации о системе» [13, с. 44], однако имитация реальной системы достоверна только отчасти.

В работе [14] указано, что нейронная сеть взаимодействует с конкретной средой, используя разные действия, в зависимости от которых агент получает числовое положительное или отрицательное вознаграждение. Также подчеркнуто, что «какое именно вознаграждение получит сеть, решается из установленных правил; если установить неоптимальный набор правил, то необходимого результата будет либо сложно достичь, либо невозможно в принципе» [14, с. 199].

Результаты исследований и экспериментов в обучении с подкреплением, а также глубоким обучением с подкреплением позволили ученым и исследователям использовать их в различных областях, где полученный результат агента превосходил человека, например, в игре GoAlphaGo [15].

Современные генеративные модели ИИ постоянно совершенствуются. Так, ChatGPT продемонстрировал большие возможности искусственного интеллекта. Ключевым требованием его развития является наличие большого объема размеченных данных (датасетов) для обучения модели, что требует значительных затрат. Обучение с подкреплением не требует большого количества данных и средств для обучения агентов, поскольку процесс обучения состоит в многократном запуске агента в среде и определении оптимальной политики.

Введем основные определения и понятия, используемые в обучении с подкреплением, рассмотрим их на примере взаимодействия агента и среды.

*Среда* является тройкой $(S, A, P)$, где $S$ – это пространство всех возможных состояний, $A$ – пространство всех действий, $P$ – функция перехода в другое состояние (Рисунок). Среда является полностью детерминированной, следовательно, указанные множества определены для всех конфигураций: $R : S \times A \to \mathbb{R}$ $(S, A, P)$; $P : S \times A \to S$.

*Агент* – обучающийся объект в обучении с подкреплением (см. Рисунок), который на каждом шаге выбирает определенное действие.

*Обучение* – процесс получения оптимальной политики агента для максимизации награды или минимизации штрафов (см. Рисунок). Существует два основных подхода в обучении: Policy-based, основанный на обучении напрямую через политику агента, и Value-based, который базируется на ценности нахождения агента в определенном состоянии.

*Политика* $\pi$ – последовательность значений состояние – действие $(s, a)$ агента, на основе которой агент выбирает действие в каждом из состояний; во время обучения политика агента изменяется на основе изучения среды и полученных наград.

*Награда* – положительное или отрицательное число, выдающееся агенту за предпринятое ранее действие: $R : S \times A \to \mathbb{R}$.

*Состояние и наблюдение* – на Рисунке показано, что на вход агенту приходит состояние *S*. В нем находится вся текущая информация о среде, агент может видеть все поле действий и расположение объектов на нем. В некоторых случаях информация о среде в со-

стоянии может быть ограничена (например, при игре в Mario информация о среде ограничена размером кадра, что обозначается как наблюдение).

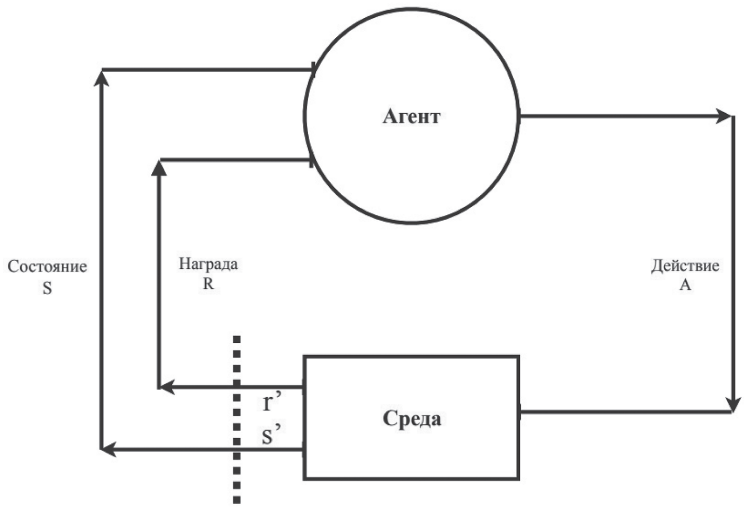


**Рисунок.** Взаимодействие агента и среды
*Источник:* рисунок выполнен авторами.

На Рисунке представлена последовательность, которая представляет процесс обучения и тренировочные окружения для обучения агентов. Агент получает состояние, в котором он находится, затем выбирает действие на основе политики, переходит в новое состояние и получает награду за выбранное ранее действие. Такой алгоритм повторяется, пока агент не дойдет до терминального состояния, из которого нельзя перейти в следующее состояние среды, то есть пока не выполнит поставленную задачу. На основе полученных наград и исследования среды строится обучение. Если агент совершает действия, приносящие штрафы или небольшие награды, значит, агент недостаточно обучился; если сумма всех наград увеличивается, значит, агент обучился достаточно.

Выбор действий определяется дилеммой исследование – эксплуатация, которая является критически важной для обучения с подкреплением. Фактически агент в RL постоянно балансирует между «крайностями»:

- Exploitation – использование текущей оптимальной политики;
- Exploration – исследование новых стратегий.

В то же время различные алгоритмы RL предлагают различные подходы к решению этой дилеммы [16]. Указанная дилемма отражает ситуацию, в которой агент руководствуется политикой, приносящей определенную награду. Если политику изменить, то награда будет намного выше, но агент не предпринимает никаких действий, поскольку ему известна актуальная политика, приносящая достаточную награду. Если агент постоянно меняет политику, перманентно исследуя среду, невозможно сформировать оптимальную политику, приносящую удовлетворительную награду. Методы value-based и policy-based включают в себя способы решения данной проблемы, тогда агент не задействован в бесконечном исследовании среды или использовании одной и той же политики.

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

Формальная постановка задачи в обучении с подкреплением задается следующим образом:

1. Наличествует агент $A$, среда $S$, начальное состояние среды $s_0$, в котором находится агент и цель, которую ему необходимо выполнить.

2. Агенту необходимо определить оптимальную последовательность действий (политику) $\pi$ для достижения поставленной цели, которая принесет максимальную награду.

3. С помощью алгоритмов обучения с подкреплением агент выбирает действия, изучая и эксплуатируя среду. После определенного количества попыток агент получит оптимальную политику.

### *Тренировочные окружения*

Исследуем тренировочные окружения с использованием обучения с подкреплением – среды для обучения агентов, которые могут сформировать политику, подходящую для защиты компьютерной сети от атак злоумышленников.

Тренировочные окружения автономных кибероперациий (Autonomous Cyber Operation Gym – ACO Gym) – это сетевые среды, которые облегчают использование автономных красных и синих агентов с целью дальнейшего укрепления сетевых систем от постоянно развивающихся атак [1]. С их помощью можно моделировать определенную ситуацию взаимодействия злоумышленника и защитника в сети [17], получая политику, с помощью которой можно эффективно минимизировать потери или предотвратить атаку.

Автоматизированный синий агент – это метод, отвечающий за защиту сетевой системы, он поддерживает ее в состоянии безопасности против группы имитируемых злоумышленников, использующих недостатки сетевой системы [1].

Автоматизированный красный агент – метод, выявляющий уязвимости сетевой системы или операционных концепций с целью минимизации инцидентов безопасности, улучшения и обеспечения надежности сетевой системы [1].

В настоящее время наличествуют различные тренировочные окружения [6; 7; 18; 19]; важно учесть при исследовании те решения, которые в большей степени отражают реальные атаки, и чьи результаты, соответственно, наиболее приближены к действительности.

Проблема тренировочных сред для решения задач информационной безопасности в контексте обучения с подкреплением изучается в трудах различных исследователей. Общие концепции рассматриваются в документе OpenAI Gym: A Toolkit for Reinforcement Learning, рассматривающем платформу для создания и тестирования алгоритмов обучения с подкреплением. В руководстве указывается, что OpenAI Gym выступает набором инструментов для исследований в сфере обучения с подкреплением, содержит перманентно увеличивающуюся коллекцию эталонных задач с единым интерфейсом и сайт, позволяющий сравнивать производительность алгоритмов [20].

В исследовании [21] рассматриваются методы симуляции для обучения управления наземным транспортом, что делает акцент на исследовании виртуальных сред, соотношения симуляции и реального мира. В статье обозначается важность сокращения разрыва реальности и моделирования, описываются методы улучшения физических характеристик систем, изучения надежных политик.

Изучаются тренировочные среды для игр. Так, анализируется применение глубокого обучения в сочетании с обучением с подкреплением на Atari 2600, формирование тренировочных сред, созданных специально для игр [22].

Обучение в многопользовательских средах рассматривается в обзорном исследовании [23], классифицирующем подходы к многопользовательскому обучению с подкреплением. Делается вывод об адаптации тренировочных сред для взаимодействия между несколькими агентами.

В обзоре [24] предложены инструменты формирования безопасных тренировочных сред для обучения с подкреплением, методы, способные предотвратить негативное поведение агентов.

Статья [25] посвящена иерархическому обучению с подкреплением, отдельно рассматривается аспект структуры наград и уровня обратной связи в осуществлении тренировочных окружений, исследуется ряд областей задач для оценки подходов к иерархическому обучению с подкреплением.

В работе [26] приводятся способы адаптации тренировочных окружений в ходе обучения на базе действий агента и его достижений, что помогает улучшить тренировочные окружения для алгоритмов обучения с подкреплением. Наличествуют и определенные проблемные точки исследований – высокая сложность построения реалистичных сред, в которых можно реализовать различные сценарии безопасности.

Для подробного анализа были выбраны тренировочные окружения CybORG и CAGE Challenge, которые представляют собой определенный аспект информационной безопасности, отражающий конкретную ситуацию взаимодействия со злоумышленником [18; 19].

Они моделируют ситуации, когда злоумышленник уже находится внутри сети. В таком случае перед защитником (синим агентом) стоит задача обнаружения и последующей деактивации злоумышленника, или красного агента, а полученная стратегия, соответственно, включает в себя действия для достижения поставленной цели.

Под процессом обучения в указанных средах понимается следующее. По аналогии с классическим обучением с подкреплением среда является тройкой $S, A, P$ (однако могут добавляться и иные множества). Пространство наблюдений $S$, доступных агенту, описывается только как доступные агенту данные. Например, информация о данных доступных хостах в сети без информации о запущенных на них процессах расширяется в процессе его обучения. Пространство действий $A$ задается разработчиком среды заранее. Так, каждое действие из пространства действий влияет на среду, следовательно, она переходит в новое состояние, а агент получает наблюдение с новой информацией о среде. Формирование наград $R$ зависит от дизайна функции наград, происходит каждый временной шаг обучения, или после достижения определенной цели. Награду можно представить в качестве числа:

- положительного (награда), тогда необходимо собрать наибольшую сумму;
- отрицательного (штраф), который должен составлять наименьшую сумму.

Таким образом, агент, постоянно доходя до терминального состояния, будет обучаться, изменяя свою политику, делая ее оптимальной.

Такое представление процесса обучения может меняться в зависимости от того, какие цели преследуют разработчики. Например, в исследованиях, посвященных тестированию на проникновение [8], пространство действий агента имеет большую размерность, а агент обучается проникновению в сеть и использованию известных ему эксплойтов при необъемной инфраструктуре сети. Тем самым для определенной задачи в информационной безопасности необходимо подобрать подходящее тренировочное окружение. Дан-

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

ные тренировочные окружения основаны на OpenAI Gymnasium API [20], которое позволяет обучить агентов и соблюсти правила обучения с подкреплением при создании своего тренировочного окружения.

Тренировочное окружение CAGE Challenge является предметом изучения для исследователей, ученых и энтузиастов по обучению синих агентов для выявления красных агентов и защиты. CAGE Challenge включает три вариации окружения. Также можно исследовать конфигурацию сред и примеры обучения агентов из таблицы лидеров. В задаче моделируется ряд сценариев безопасности для облегчения разработки автономных синих агентов. Автономные красные агенты разрабатываются для каждого конкретного сценария. Ключевая цель CAGE Challenge – разработка синих агентов с эффективной стратегией, направленной на защиту от атак красных агентов.

Среда в тренировочном окружении CAGE Challenge неизменна, нельзя создать свою конфигурацию сети или параметры агента и среды, поскольку в каждой из задач уже сформирована собственная сеть и определены пространства действий и состояний. Затем на базе данных конфигураций проводятся соревнования по созданию эффективных синих агентов. CAGE Challenge основан на фреймворке CybORG [18], в котором можно создавать собственные конфигурации и прописывать необходимые для поставленных целей условия среды.

Каждое соревнование проводилось в новой версии задачи: CAGE Challenge 1, CAGE Challenge 2 и CAGE Challenge 3 соответственно. Каждое из них различается условиями, устройством сети и другими параметрами. В соревнованиях 1 и 2 среда представляла собой компьютерную сеть, разделенную на три сегмента по важности устройств и их содержимого, третье соревнование вместо указанной ранее сети использовало беспилотные летательные аппараты на ОС Linux с известной для всех агентов уязвимостью. Это показывает возможности представления различных задач с помощью тренировочных окружений.

В CAGE Challenge 1 и 2 наличествует несколько сегментов сети, разделенной межсетевыми экранами, в каждом из которых находится определенное количество компьютеров, выполняющих различные функции. В первой подсети содержатся пользовательские компьютеры, во второй – компьютеры, поддерживающие работу пользовательских компьютеров. Здесь начинает обучение синий агент. В третьей подсети – серверы критической инфраструктуры, отвечающие за работу всей инфраструктуры предприятия. Красный и синий агенты могут взаимодействовать с элементами в рамках доступных действий из пространства действий, разработанного авторами.

Таким образом, представление сети в CAGE Challenge 2 имитирует реальную компьютерную сеть предприятия, а начальной точкой задачи является момент времени, когда злоумышленник в лице красного агента уже проник в сеть и получил доступ на одном из пользовательских ПК в первой подсети. В CAGE Challenge 2 присутствуют несколько красных агентов, их цель – добраться до рабочего сервера и повлиять на сетевые сервисы, чтобы нанести ущерб. Каждый агент действует по-разному с учетом имеющихся паттернов поведения. Задача синего агента – минимизировать воздействие злоумышленника, сохраняя функциональность сети. Для каждой из этих целей существуют отдельные пространства действий, которые красный или синий агент могут предпринимать. Среда является частично наблюдаемой и стохастической, и в нее входят следующие параметры:

- $I$ – набор проиндексированных агентов;
- $S$ – набор состояний;

• *A* – набор пространства действий;
• *O* – набор наблюдений для каждого агента;
• *T* – вероятности перехода между состояниями;
• *R* – функция вознаграждения.

Таким образом, среда в CAGE Challenge 2 описывается кортежем из множеств (*I*, *S*, *A*, *O*, *T*, *R*) [18].

В CAGE Challenge 2 наличествуют три уникальных красных агента с готовыми наборами правил поведения:

• B-Line – осведомлен о сети, выбирает действия, направленные только на воздействие на сервера за наименьшее количество шагов;

• Meander – не имеет представления о сети, действия направлены на изучение сети и получение привилегированного доступа к узлам сети;

• Sleep – существует для того, чтобы его обнаружил синий агент, осуществляя проверку на наличие угрозы в системе [18]. Sleep позволяет проверить универсальность полученной стратегии синего агента против красных или сконцентрироваться на максимально эффективном противостоянии определенному красному агенту.

Пространства действий синего и красного агентов различные, за исключением действия Sleep, которое присутствует в обоих пространствах. Действие Sleep выполняет своеобразный пропуск хода – не влияя на среду и пропуская временной шаг.

У синего агента пространство действий постоянно, и в свой ход он может выбрать любое из них, в то время как у красного агента пространство действий увеличивается пропорционально тому, какой объем сети ему доступен. Рассмотрим подробнее пространство действий синего агента. Все его действия могут быть сгруппированы в три различные категории по признаку их направленности:

• обман – на создание ловушек, которые будут замедлять продвижение красного агента;

• разведка – сбор данных о серверах и состоянии сети;

• восстановление – необходимо для удаления последствий действий красного агента на компьютерах в сети [18].

Пространство действий красного агента также дифференцируется на три группы:

• разведка – сбор сведений, красный агент получает информацию о службах и уязвимостях в них;

• эксплуатация – использование ранее найденных уязвимостей для повышения привилегий;

• влияние – внесение неисправности в работу сети.

Техническая реализация описанного выше тренировочного окружения базируется на CybORG – решении, состоящем из симулятора и эмулятора, позволяющем провести обучение и проверку получившейся политики агента в условиях, приближенным к реальным.

*Симулятор сети* – тип сети, основанный на описании конфигурации и составляющих сети в конфигурационном файле. Указанная конфигурация симулируется, формируя среду для обучения агента. Полученные результаты являются менее точными, но менее трудоемкими и ресурсоемкими.

*Эмулятор сети* – тип сети, базирующийся на средствах виртуализации, что позволяет включить в среду больший объем параметров. Полученные результаты приближены к реальной инфраструктуре, поэтому являются наиболее точными, но более ресурсоемкими.

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

Данная дифференциация необходима, поскольку существует «разрыв в реальности» [19] между эффективностью агентов, прошедших обучение только в симуляторах или эмуляторах. Агент при обучении в симуляторе не сталкивается с некоторыми состояниями или действиями, учтенными в эмуляторе, который имеет больше параметров в работе операционной системы, чем симулятор, использующий текстовый файл для симуляции операционной системы и процессов в ней. Соответственно, отсутствие такой проверки может привести к ситуации неэффективности агента в реальных сетях.

CAGE Challenge, основанный на CybORG, также использует доступные инструменты симуляции и эмуляции. Задачи CAGE Challenge 1, 2 и 3 являются отдельной сущностью в CybORG, называющейся сценарием, определяющим базовые положения для дальнейшего процесса обучения:

- информация о сетевых устройствах в компьютерной сети;
- сети и подсети;
- пространство действий красных и синих агентов;
- количество красных и синих агентов.

Описать сценарий возможно с использованием YAML – единого формата для конфигурации, использующего написанный сценарий для симуляции и эмуляции. Симулятор работает на основе конечного автомата, в котором реализован весь функционал обучения с подкреплением (наблюдение, состояния, действия и др.), а полученные значения передаются агенту для продолжения обучения или запуска симулятора заново.

Для работы эмулятора необходима реальная инфраструктура или определенный инструмент виртуализации. Так, Docker. CybORG использует AmazonWebServices (AWS), создавая виртуальные машины, сети, подсети и маршрутизацию по готовым скриптам в облаке. Затем в получившуюся сеть передаются действия агента, а новое состояние сети возвращается агенту. Использование IaaS-подхода, как и подхода с локальной виртуализацией, при наличии достаточного ресурса мощности обеспечивает масштабируемость и гибкость тренировочного окружения.

Вопрос о приближенности тренировочных окружений к реальности можно рассмотреть с точки зрения получаемых результатов от симулятора и эмулятора. Существует «разрыв в реальности» между эффективностью полученных политик агентов; для его нивелирования необходимо увеличивать количество параметров, доступных агенту для наблюдения в среде.

В этом случае нужно ориентироваться не на увеличение размеров доступных подсетей и содержащихся в них хостов, а на количество параметров хостов: запущенные процессы, количество пользователей и доступные подсети. Чем больше таких параметров хостов в симуляторе, тем точнее и эффективнее полученная политика агента. В тренировочном окружении CybORG не приведено шаблонного сценария для проверки количества указанных параметров, однако в задачи CAGE Challenge предоставлен файл сценария, после изучения которого можно понять, что количество пользователей на хостах равно двум, запущенных процессов – шести, а запущенных сервисов – трем. Тогда увеличение на несколько единиц будет очень значительным. Для изменения параметров необходимо изменить логику вычисления размерности пространства наблюдений агента, чтобы исключить ошибку в расчетах.

Выведение основных компонентов тренировочных окружений и их систематизацию можно представить в виде таблицы.

*Таблица*

**Компонент тренировочных окружений**

| Компонент | Степень важности | Уровень реализации для пользователя |
|---|---|---|
| Наличие эмулятора | Средняя | Доступен |
| Наличие сценария | Высокая | Доступен |
| Различное поведение красных агентов | Средняя | Недоступен |
| Постановка цели синего агента | Низкая | Недоступно |
| Добавление собственного пространства действий | Средняя | Доступно |

*«Степень важности»* может быть низкой, средней и высокой. Компонент означает критичность наличия того или иного компонента в тренировочном окружении. Данная величина означает необходимость присутствия определенного аспекта в тренировочном окружении.

Компонент *«уровень реализации»* выступает представлением меры доступности аспекта для его изменения пользователем платформы, демонстрирует необходимость доступности пользователю выбранного аспекта для модификации.

Компонент *«наличие сценария»* является значимым в тренировочном окружении, а «высокая важность» позволяет внедрять и реализовывать обучение агента в собственной инфраструктуре.

*Уровень реализации* принимает значение «доступен пользователю», поскольку данный функционал необходим для обеспечения возможности внесения изменений в сценарий для получения политики для различных инфраструктур. Сценарии формируют контекст для принятия решений агентом. Наличие конкретных сценариев способствует обучению агентов на конкретных примерах, что приоритетно для эффективного обучения.

Компонент *«наличие эмулятора»* выделяет необходимость наличия функционала эмулятора в тренировочном окружении. Эмуляторы используют виртуальные машины с большим количеством параметров в наблюдениях агентов, тем самым результаты обучения получаются более приближенными к среде. Уровень реализации «доступен пользователю» необходим для собственной настройки пользователем облачного провайдера, а также тарифа на виртуальные машины.

Компонент *«степень важности»* установлен в значении «средняя» из-за трудности реализации доступности платформ IaaS, а также различий в синтаксисе скриптов автоматизированного развертывания инфраструктуры в них.

Компонент *«различное поведение красных агентов»* имеет свойства «средняя важность» и «недоступен пользователю» по причине наличия в тренировочном окружении конечного числа дифференцирующих действий красного агента; стандартные красные агенты представляют все возможные комбинации действий. Разнообразие поведения агентов делает обучение более реалистичным, усложняя задачу. Детальное добавление различного поведения агентов определенно сопряжено с наличием такого функционала, как модель нарушителя, а также с увеличением пространства действий агентов.

Компонент *«постановка цели синего агента»* исследуется с точки зрения возможности пользователю задать любую цель агенту. В обучении с подкреплением это затратный

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

процесс, сопряженный с редактированием функции награды, поэтому ему присвоены низкая важность и недоступность для пользователя, хотя это может ограничить гибкость.

Компонент *«добавление собственного пространства действий»* означает, что пользователю доступно изменение пространства действий агентов. Это условие достижимо, как и во фреймворке CybORG:

• с помощью модификации исходного кода тренировочного окружения (простой способ, но могут быть зависимости в коде);

• создания интерфейса для добавления новых элементов пространства действий (сложный способ).

Возможность добавлять собственное пространство действий помогает пользователям адаптировать окружение для своих потребностей и экспериментировать с разными стратегиями, что обеспечивает появление инноваций.

Важным является наличие сценария в CAGE Challenge, поскольку это выступает базисом для обучения агентов в меняющихся условиях. Уровень реализации для пользователя отмечен доступностью и настраиваемостью. Это делает CAGE Challenge результативным тренировочным окружением для исследователей в сфере обучения с подкреплением.

CybORG включает сценарии, относящиеся к кибербезопасности и моделированию атак и защитных действий. Сценарии адаптируются под конкретные задачи, что облегчает исследование разных аспектов киберзащиты. Пользователи могут работать с имеющимися сценариями или формировать новые, и это позволяет им настраивать обучение агентов под специальные задачи. Цели синего агента CAGE Challenge и CybORG четко определены, но в последнем случае сосредоточены на защите от атак и обеспечении безопасности системы, поэтому доступ к ним ограничен.

### *Результаты и их обсуждение*

В рамках исследования устройства тренировочных окружений и их основных аспектов рассмотрены среды CAGE Challenge и CybORG. По итогам исследования выдвинуты следующие основные выводы.

Симулятор и эмулятор сети должны соответствовать правилам обучения с подкреплением, а именно: принимать на вход действие агента, после чего изменять состояние среды и отдавать получившееся состояние или наблюдение и награду за предыдущий шаг. Это позволит применять различные методы и алгоритмы обучения с подкреплением для обучения агентов и получения наиболее оптимальной политики.

Анализ характеристик тренировочных окружений, основанных на обучении с подкреплением, позволяет определить ключевые нюансы их функционирования:

• *изменение и развитие среды* – тренировочные окружения характеризуются подвижной динамикой, что определяет уровень сложности задач агента; динамические среды с трансформирующимся от функционирования агента состоянием нуждаются в углубленном изучении, сложной адаптации и стратегии в отличие от статических сред;

• *награды и штрафы* – структура наград и штрафов в среде является приоритетной в обучении с подкреплением; выделение наград, их точный ранг совершенствуют процесс обучения, тогда как неопределенные награды, представляемые в хаотичном порядке, способствуют неверному поведению агента;

• *уровень задач* – степень сложности задач имеет прямую корреляцию с обучаемостью агента; переходя от уровня к уровню, от простых задач к сложным, агент развивает свои

навыки; принцип постепенности обучения позволяет фиксировать успешный опыт и применять его на более сложных уровнях;

• *обратная связь* – для осуществления своей политики и адаптации агенту важны скорость и качество полученной обратной связи; быстрая и четкая обратная связь способствует ускорению процесса обучения;

• *настройка окружения* – гибкость в настройке окружения способствует адаптированию его пользователями, что приводит индивидуализированному взаимодействию и результативному обучению; добавление новых сценариев, смена параметров, влияние на поведение агентов – всё это расширяет пространство для действия;

• *взаимодействие с агентами* – в многопользовательских средах наличествует ряд агентов, необходимо контролировать взаимодействие между ними; агентам необходимо учитывать действия других агентов, что требует переориентации и функционирования на более высоком уровне;

• *степень случайности* – наличие случайных компонентов в среде делает процесс обучения непредсказуемым и затрудненным. Агент должен уметь действовать в условиях неопределенности и формировать стратегии исходя из условий трансформации среды.

На основе системного изучения работ [20–26], детального анализа платформ CybORG и CAGE Challenge, а также полученной таблицы можно вывести список ключевых критериев эффективных тренировочных окружений, а именно:

• ***наличие сценария.*** Основная инфраструктура вносится в тренировочное окружение путем использования функционала добавления сценария, что позволяет быстро выполнять процесс внедрения новой инфраструктуры и, соответственно, тренировочное окружение становится более гибким;

• ***наличие эмулятора и симулятора.*** Симулятор необходим для запуска обучения, не используя при этом большее количество ресурсов, в то же время для проверки полученной политики в условиях, приближенных к реальным сетям, необходим эмулятор, требующий больших вычислительных ресурсов;

• ***дифференциация тренировочных окружений по реалистичности среды.*** Различные тренировочные окружения используются для различных задач. Так, тренировочные окружения со слабой «реалистичностью» используются для проверки алгоритмов RL, а окружения, приближенные к реальности, могут использоваться в качестве инструментов по безопасности;

• ***изменение параметров красного и синего агентов.*** Изменение параметров красного и синего агентов, таких как их цели и пространство действий, используется для более точной настройки обучения под собственную инфраструктуру сети, поэтому этот критерий является ключевым;

• ***конфигурация сценария путем изменения единого конфиг-файла.*** Конфигурация параметров должна содержаться в единых конфигурационных файлах, например, YAML или XML, позволяя быстро вносить изменения в функционал тренировочного окружения.

Представленный список критериев позволяет специалистам по безопасности или исследователям подойти к вопросу выбора тренировочного окружения с точки зрения использования вышеизложенных критериев. Ранее из-за новизны этой области выбор тренировочного окружения мог производиться только на основе внутренней экспертизы компании. Дополнительно полученные критерии можно использовать как основу для создания собственного тренировочного окружения.

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

Таким образом, научная новизна исследования заключается в следующих пунктах:

***1) формализация требований к оценке автономных сред тренировочных окружений (АСТО)*** – созданы ранее не существовавшие требования к АСТО;

***2) комплексность исследования*** – учтены различные задачи информационной безопасности, которые можно воспроизвести в тренировочном окружении, проанализированы текущие работы по исследованию применения RL в информационной безопасности.

### *Заключение*

Анализ аспектов тренировочных окружений показывает, что наличие функционала добавления сценария, а также возможность добавления собственного пространства действий являются ключевыми факторами гибкого тренировочного окружения. В то же время ограничения в настройке поведения агентов и постановке целей могут повлиять на гибкость и эффективность обучения. Выбор окружения должен основываться на конкретных задачах и целях исследования, учитывая представленные критерии:

• результаты агента, полученные в симуляции, необходимо дополнительно проверить на корректность и валидность при защите реальных сетевых инфраструктур с помощью эмулятора, поскольку существует «разрыв в реальности», показывающий, что не все стратегии, полученные путем обучения агента в симуляции, могут применяться для защиты сетевой инфраструктуры из-за различия между количеством параметров в среде;

• тренировочные окружения дифференцируются по их характеристикам: тип среды (симуляция, реальный мир, игры), структура наград, степень сложности и динамика взаимодействия;

• необходима возможность задать поведение красного агента для реализации различных сценариев безопасности;

• при создании тренировочного окружения необходимо учитывать цель синего агента – это определяет наличие или отсутствие других агентов, а также размерность пространства действий и состояний;

• для работы с тренировочным окружением необходимо использовать конфигурацию в виде YAML-файлов – такой подход несет в себе комфорт работы с известным и читаемым форматом написания конфигурации и быстрого внесения изменений в среду;

• проблему приближенности тренировочных окружений к реальности необходимо рассматривать одновременно с вопросом о создании и внедрении собственного сценария или формирования собственного тренировочного окружения, в котором учитывается встроенное добавление данных параметров для увеличения размерности пространства наблюдений, что уменьшает «разрыв в реальности».

По итогам исследования получен следующий вывод: автоматизированная информационная защита является инновационным и необходимым подходом к безопасности в современном мире. Обучение с подкреплением и глубокое обучение с подкреплением, которые дают возможность обучения без предварительной подготовки и без сбора и маркирования данных, являются ведущими методами создания автономных красных и синих агентов и тренировочных окружений. Для обучения агентов необходимы автономные тренировочные окружения, отражающие определенный аспект информационной безопасности. Разработчик тренировочного окружения создает окружение с возможностью создания собственных сценариев, что позволит реализовать свойства масштабируемости, гибкости и удобства использования.

Тренировочные окружения функционируют в разных средах (симуляция, реальный мир, игры) и в разных областях (финансы, здравоохранение, робототехника и др.) и должны быть построены с учетом принципов работы обучения с подкреплением, чтобы происходил процесс обучения агента.

Выделены необходимые значимые свойства автономного тренировочного окружения, использующего обучение с подкреплением. Основными факторами эффективного обучения с подкреплением выступают наличие сценариев и возможность добавления собственного пространства действий. В то же время ограничения в настройке поведения агентов и постановке целей могут повлиять на гибкость и эффективность обучения. Выбор окружения должен основываться на конкретных задачах и целях исследования с учетом параметров каждого компонента.

Получены критерии для оценки автономных сред тренировочных окружений (АСТО) для задач информационной безопасности. Критерии основаны на изучении текущих работ в области АИЗ и фреймворков CAGE Challenge и CybORG. Критерии содержат в себе важные функции тренировочных окружений, поэтому они позволяют задать требования при разработке новых АСТО и на их основе осуществлять выбор подходящих тренировочных окружений.

Функционирование тренировочных окружений в обучении с подкреплением определяется рядом факторов: динамика среды, возможности настройки, структура наград, качество обратной связи. Понимание данных особенностей способствует результативной адаптации окружения под определенные задачи и цели и эффективному обучению агентов.

**Литература**

1. *Vyas S., Hannay J., Bolton A., Burnap P.* Automated cyber defence: A review // arXiv. 2023. DOI: https://doi.org/10.48550/arXiv.2303.04926

2. *Zihao Wang, Kar-Wai Fok, Vrizlynn L.L. Thing.* Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study // Computers Security. 2022. Vol. 113. Article no. 102542. DOI: https://doi.org/10.1016/j.cose.2021.102542

3. *Hammar K., Stadler R.* Intrusion prevention through optimal stopping // IEEE Transactions on Network and Service Management. 2022. Vol. 19. No. 3. Pp. 2333–2348. DOI: https://doi.org/10.1109/tnsm.2022.3176781

4. *Doya K.* Reinforcement learning in continuous time and space // Neural Computation. 2000. Vol. 12. No. 1. Pp. 219–245. DOI: 10.1162/089976600300015961

5. *Oh S.H., Kim J., Nah J.H., Park J.* Employing Deep Reinforcement Learning to Cyber-Attack Simulation for Enhancing Cybersecurity // Electronics. 2024. Vol. 13. No. 3. Article no. 555. DOI: https://doi.org/10.3390/electronics13030555

6. *Molina-Markham A., Miniter C., Becky P., Ridley A.* Network environment design for autonomous cyberdefense //arXiv. 2021. DOI: https://doi.org/10.48550/arXiv.2103.07583

7. *Andrew A., Spillard S., Collyer J., Dhir N.* Developing optimal causal cyber-defence agents via cyber security simulation // arXiv. 2022. DOI: https://arxiv.org/abs/2207.12355

8. *Nguyen T.T., Reddi V.J.* Deep reinforcement learning for cyber security // IEEE Transactions on Neural Networks and Learning Systems. 2021. Vol. 34. No. 8. P. 3779–3795. DOI: 10.1109/TNNLS.2021.3121870

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

9. *Del Verme M., Sommervoll A.A., Erdődi L., Totaro S., Zennaro F.M.* SQL injections and reinforcement learning: An empirical evaluation of the role of action structure // Tuveri N., Michalas A., Brumley B.B. (Eds) Secure IT Systems: 26th Nordic Conference, NordSec 2021, Virtual Event, November 29–30, 2021, Proceedings 26. Series: Lecture Notes in Computer Science. Vol. 13115. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-91625-1_6
10. *Sewak M., Sahay S.K., Rathore H.* Deep reinforcement learning for cybersecurity threat detection and protection: A review // Krishnan R., Rao H.R., Sahay S.K., Samtani S., Zhao Z. (Eds) Secure Knowledge Management in the Artificial Intelligence Era. SKM 2021. Series: Communications in Computer and Information Science. Vol. 1549. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-97532-6_4
11. *Zhou S., Liu J., Hou D., Zhong X., Zhang Y.* Autonomous penetration testing based on improved deep q-network // Applied Sciences. 2021. Vol. 11. No. 19. Article no. 8823. DOI: https://doi.org/10.3390/app11198823
12. *Кошманова Н.П., Трифонов Д.С., Павловский В.Е.* Управление манипулятором с помощью обучения с подкреплением // Нелинейная динамика. 2012. Т. 8. № 4. С. 689–704. EDN PKTALT.
13. *Орлова Е.В.* Обучение с подкреплением как технология искусственного интеллекта для решения социально-экономических задач: оценка производительности алгоритмов // π-Economy. 2023. Т. 16. № 5. С. 38–50. DOI: 10.18721/JE.16503. EDN OHKJKP.
14. *Килин Г.А., Кавалеров Б.В., Ждановский Е.О., Бахирев И.В., Опарин Д.А.* Программный комплекс для реализации обучения с подкреплением // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. 2018. № 27. С. 195–208. EDN YGIGFV.
15. *Hölldobler S., Möhle S., Tigunova A.* Lessons Learned from AlphaGo // Holldobler S., Malikov A., Wernhard C. (Eds) YSIP2 Proceedings of the Second Young Scientists International Workshop on Trends in Information Processing, Dombai, Russian Federation, May 16–20, 2017. Pp. 92–101. URL: https://www.sibyllemoehle.net/images/pdf/HoelldoblerMoehleTigunova-YSIP17.pdf (дата обращения: 12.04.2025).
16. *Sutton R., Barto A.* Reinforcement learning: An Introduction. Cambridge, MA : MIT Press, 1998. 322 p. ISBN 0262193981.
17. *Yoo J.D., Park E., Lee G., Ahn M.K., Kim D., Seo S., Kim H.K.* Cyber attack and defense emulation agents // Applied Sciences. 2020. Vol. 10. No. 6. Article no. 2140. DOI: https://doi.org/10.3390/app10062140
18. *Kiely M., Bowman D., Standen M., Moir C.* On Autonomous Agents in a Cyber Defence Environment // arXiv, 2023. DOI: https://doi.org/10.48550/arXiv.2309.07388
19. *Standen M., Lucas M., Bowman D., Richer T.J., Kim J., Marriott D.* Cyborg: A gym for the development of autonomous cyber agents // arXiv. 2021. DOI: https://doi.org/10.48550/arXiv.2108.09118
20. Brockman G., Cheung V., Pettersson L., Schneider J., Schulman J., Tang J., Zaremba W. OpenAI Gym // arXiv. 2016. DOI: https://doi.org/10.48550/arXiv.1606.01540
21. *Tan J., Zhang T., Coumans E., Iscen A., Bai Y., Hafner D., Bohez S., Vanhoucke V.* Sim-to-Real: Learning Agile Locomotor Policies for Quadruped Robots // arXiv. 2018. DOI: https://doi.org/10.48550/arXiv.1804.10332
22. *Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M.* Playing Atari with Deep Reinforcement Learning // arXiv. 2013. DOI: https://doi.org/10.48550/arXiv.1312.5602
23. *Dom H., Mohapatra P.* Multi-Agent Reinforcement Learning: A Review // arXiv. 2023. DOI: https://doi.org/10.48550/arXiv.2312.10256

24. *Zhao W., He T., Chen R., Wei T., Liu C.* State-wise Safe Reinforcement Learning: A Survey // arXiv. 2023. DOI: https://doi.org/10.48550/arXiv.2302.03122
25. *Pateria S., Subagdja B., Tan A., Quek C.* Hierarchical Reinforcement Learning: A Comprehensive Survey // ACM Computing Surveys. 2021. Vol. 54. No. 5. Pp. 1–35. URL: https://ink.library.smu.edu.sg/sis_research/6047 (дата обращения: 12.04.2025).
26. *Wang S., Gao R., Han R., Chen S., Li C., Hao Q.* Adaptive Environment Modeling Based Reinforcement Learning for Collision Avoidance in Complex Scenes // arXiv. 2022. DOI: https://doi.org/10.48550/arXiv.2203.07709

**References**

1. Vyas S., Hannay J., Bolton A., Burnap P. (2023) Automated cyber defence: A review. *arXiv*. DOI: https://doi.org/10.48550/arXiv.2303.04926
2. Zihao Wang, Kar-Wai Fok, Vrizlynn L.L. Thing. () Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study. *Computers Security.* 2022. Vol. 113. Article no. 102542. DOI: https://doi.org/10.1016/j.cose.2021.102542
3. Hammar K., Stadler R. (2022) Intrusion prevention through optimal stopping. In: *IEEE Transactions on Network and Service Management*. Vol. 19. No. 3. Pp. 2333–2348. DOI: https://doi.org/10.1109/tnsm.2022.3176781
4. Doya K. (2000) Reinforcement learning in continuous time and space. *Neural Computation*. 2000. Vol. 12. No. 1. Pp. 219–245. DOI: 10.1162/089976600300015961
5. Oh S.H., Kim J., Nah J.H., Park J. (2024) Employing Deep Reinforcement Learning to Cyber-Attack Simulation for Enhancing Cybersecurity. *Electronics*. 2024. Vol. 13. No. 3. Article no. 555. DOI: https://doi.org/10.3390/electronics13030555
6. Molina-Markham A., Miniter C., Becky P., Ridley A. (2021) Network environment design for autonomous cyberdefense. *arXiv*. DOI: https://doi.org/10.48550/arXiv.2103.07583
7. Andrew A., Spillard S., Collyer J., Dhir N. (2022) Developing optimal causal cyber-defence agents via cyber security simulation. *arXiv*. DOI: https://arxiv.org/abs/2207.12355
8. Nguyen T.T., Reddi V.J. (2021) Deep reinforcement learning for cyber security. In: *IEEE Transactions on Neural Networks and Learning Systems.* Vol. 34. No. 8. P. 3779–3795. DOI: 10.1109/TNNLS.2021.3121870
9. Del Verme M., Sommervoll A.A., Erdődi L., Totaro S., Zennaro F.M. (2021) SQL injections and reinforcement learning: An empirical evaluation of the role of action structure. In: Tuveri N., Michalas A., Brumley B.B. (Eds) *Secure IT Systems: 26th Nordic Conference, NordSec 2021, Virtual Event, November 29–30, 2021, Proceedings 26.* Series: Lecture Notes in Computer Science. Vol. 13115. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-91625-1_6
10. Sewak M., Sahay S.K., Rathore H. (2021) Deep reinforcement learning for cybersecurity threat detection and protection: A review. In: Krishnan R., Rao H.R., Sahay S.K., Samtani S., Zhao Z. (Eds) *Secure Knowledge Management in the Artificial Intelligence Era. SKM 2021.* Series: Communications in Computer and Information Science. Vol. 1549. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-97532-6_4
11. Zhou S., Liu J., Hou D., Zhong X., Zhang Y. (2021) Autonomous penetration testing based on improved deep q-network. *Applied Sciences.* Vol. 11. No. 19. Article no. 8823. DOI: https://doi.org/10.3390/app11198823

Вестник Российского нового университета
Серия «Сложные системы: модели, анализ и управление». 2025. № 3

12. Koshmanova N.P., Trifonov D.S., Pavlovsky V.E. (2012) Reinforcement learning for manipulator control. *Russian Journal of Nonlinear Dynamics*. Vol. 8. No. 4. Pp. 689–704. (In Russian).
13. Orlova E.V. (2023) Reinforcement Learning as an Artificial Intelligence Technology to Solve Socio-Economic Problems: algorithms performance assessment. *π-Economy.* Vol. 16. No. 5. Pp. 38–50. DOI: 10.18721/JE.16503 (In Russian).
14. Kilin G.A., Kavalerov B.V., Zhdanovsky E.O., Bakhirev I.V., Oparin D.A. (2018) Software package for implementing reinforcement learning. *Bulletin of Perm National Research Polytechnic University. Electrotechnics, information technologies, control systems*. No. 27. Pp. 195–208. (In Russian).
15. Hölldobler S., Möhle S., Tigunova A. (2017) Lessons Learned from AlphaGo. In: Holldobler S., Malikov A., Wernhard C. (Eds) *YSIP2 Proceedings of the Second Young Scientists International Workshop on Trends in Information Processing*. Dombai, Russian Federation, May 16–20, 2017. Pp. 92–101. URL: https://www.sibyllemoehle.net/images/pdf/HoelldoblerMoehleTigunova-YSIP17.pdf (accessed 12.04.2025).
16. Sutton R., Barto A. (1998) *Reinforcement learning: An Introduction*. Cambridge, MA : MIT Press, 1998. 322 p. ISBN 0262193981.
17. Yoo J.D., Park E., Lee G., Ahn M.K., Kim D., Seo S., Kim H.K. (2020) Cyber attack and defense emulation agents. *Applied Sciences* Vol. 10. No. 6. Article no. 2140. DOI: https://doi.org/10.3390/app10062140
18. Kiely M., Bowman D., Standen M., Moir C. (2023) On Autonomous Agents in a Cyber Defence Environment. *arXiv*. DOI: https://doi.org/10.48550/arXiv.2309.07388
19. Standen M., Lucas M., Bowman D., Richer T.J., Kim J., Marriott D. (2021) Cyborg: A gym for the development of autonomous cyber agents. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2108.09118
20. Brockman G., Cheung V., Pettersson L., Schneider J., Schulman J., Tang J., Zaremba W. (2016) OpenAI Gym. *arXiv.* DOI: https://doi.org/10.48550/arXiv.1606.01540
21. Tan J., Zhang T., Coumans E., Iscen A., Bai Y., Hafner D., Bohez S., Vanhoucke V. (2018) Sim-to-Real: Learning Agile Locomotor Policies for Quadruped Robots. *arXiv.* DOI: https://doi.org/10.48550/arXiv.1804.10332
22. Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M. (2013) Playing Atari with Deep Reinforcement Learning. *arXiv.* DOI: https://doi.org/10.48550/arXiv.1312.5602
23. Dom H., Mohapatra P. (2023) Multi-Agent Reinforcement Learning: A Review. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2312.10256
24. Zhao W., He T., Chen R., Wei T., Liu C. (2023) Safe Reinforcement Learning: A Survey. *arXiv.* 2023. DOI: https://doi.org/10.48550/arXiv.2302.03122
25. Pateria S., Subagdja B., Tan A., Quek C. (2021) Hierarchical Reinforcement Learning: A Comprehensive Survey. *ACM Computing Surveys.* Vol. 54. No. 5. Pp. 1–35. URL: https://ink.library.smu.edu.sg/sis_research/6047 (accessed 12.04.2025).
26. Wang S., Gao R., Han R., Chen S., Li C., Hao Q. (2022) Adaptive Environment Modeling Based Reinforcement Learning for Collision Avoidance in Complex Scenes. *arXiv.* DOI: https://doi.org/10.48550/arXiv.2203.07709

| | |
|---|---|
| Поступила в редакцию: 28.04.2025 | Received: 28.04.2025 |
| Поступила после рецензирования: 23.05.2025 | Revised: 23.05.2025 |
| Принята к публикации: 11.06.2025 | Accepted: 11.06.2025 |