Список использованных источников

1. Система управления электродинамическим ускорителем с вычислением параметров движения в реальном времени / А. В. Пияков, К. И. Сухачев, А. С. Дорофеев, В. А. Бандяев // Труды МАИ. – 2022. – № 124. – DOI 10.34759/trd-2022-124-20. – EDN QAXCKU.

2. Реализация отказоустойчивой межкристальной связив системах космической научной аппаратуры, на базе нескольких программируемых логических интегральных схем / К. Е. Воронов, К. И. Сухачев, Д. А. Шестаков, А. А. Артюшин // Ракетно-космическое приборостроение и информационные системы. – 2022. – Т. 9, № 3. – С. 57-64. – DOI 10.30894/issn2409-0239.2022.9.3.57.64. – EDN WVAAJN.

3. Разработка кольцевой полудуплексной сети для обмена данными между устройствами в научной космической аппаратуре / К. И. Сухачев, Д. П. Григорьев, Д. А. Шестаков [и др.] // Вестник Рязанского государственного радиотехнического университета. – 2023. – № 84. – С. 34-45. – DOI 10.21667/1995-4565-2023-84-34-45. – EDN AHVMXO.

4. Свидетельство о государственной регистрации программы для ЭВМ № 2023619915 Российская Федерация. Синтезируемый в базисе ПЛИС комплект модулей, реализующих сетевой протокол связи IL NET: № 2023618627: заявл. 03.05.2023: опубл. 17.05.2023 / К. И. Сухачев, Д. П. Григорьев, А. А. Артюшин, Д. А. Шестаков; заявитель федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева». – EDN XAYVKW.

Краснов Вячеслав Валерьевич, студент гр. 3465-110303D, krasnov.vv@ssau.ru.
Телегин Алексей Михайлович, к.ф-м.н. доцент каф. РЭС, telegin.am@ssau.ru.

УДК 004.056.5:378.3

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ МЕТОДОВ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ ПРИ МОДЕЛИРОВАНИИ АТАК НА СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

В. В. Подтопельный
Калининградский государственный технический университет,
г. Калининград

С развитием систем искусственного интеллекта (ИИ), включая генеративно-состязательные сети (GAN), возрастает их уязвимость к многоэтапным кибератакам, направленным на компрометацию функциональности [1, 2]. Для формального описания поведения атакующих

в таких сценариях широко применяются марковские процессы принятия решений (МППР), которые позволяют представить действия злоумышленника как последовательность оптимальных шагов [3]. Эффективность подобных моделей во многом определяется выбором алгоритма их решения [4]. В настоящее время для решения задач кибербезопасности активно используются два метода динамического программирования: Value Iteration (VI), основанный на итеративном обновлении функции ценности до достижения сходимости, и Policy Iteration (PI), предполагающий чередование этапов оценки и улучшения текущей стратегии [5, 6]. Методы динамического программирования гарантируют сходимость к оптимальному решению в полностью определённых моделях МППР, используемых в решении задач кибербезопасности ИИ. При этом требуется проводить дальнейшее изучение особенностей их применения, особенно в свете различий в вычислительных характеристиках при соблюдении требований стандартов в области ИБ (MITRE ATLAS и др.) [1, 7]. Настоящая работа направлена на анализ методов динамического программирования (VI и PI) с учётом скорости сходимости, полноты моделирования (способности учитывать все состояния, действия и переходы) и вычислительной эффективности в условиях моделирования атак на ИИ-системы.

Для проведения анализа разработана МППР-модель, описывающая атаку на GAN-систему [8]. Модель включает пять состояний: разведка (S11), подготовка (S2), доступ (S3), воздействие (S4) и блокировка (S5). Действия атакующего представлены тремя категориями: нелегитимная эксплуатация уязвимостей (D), легитимная эксплуатация уязвимостей (C) и сброс состояния (R). Параметры модели определены с коэффициентом дисконтирования γ=0.9, а матрицы переходов и вознаграждений синтезированы на основе данных реальных атак, таких как эксплуатация уязвимостей в API ИИ-систем [9]. Для VI начальные значения функции ценности установлены как $V_0(s)=0$, а критерий остановки задан как $\max_s \|Vk+1(s)-Vk(s)\|<0.001$. Для метода PI начальная стратегия определена как $\pi_0(s)=C$, с критерием остановки в виде изменения политики менее чем на 1% за итерацию. Оценка методов проводилась по трём метрикам: время сходимости (в миллисекундах), использование памяти (в мегабайтах) и полнота моделирования (в процентах учёта переходов и действий) [6]. При исследовании использовались три сценария тестирования. Базовый сценарий с пятью состояниями был направлен на сравнение скорости и точности методов. Масштабируемый сценарий с числом состояний от 20 до 100, сгенерированных методом Монте-Карло, позволил оценить производительность при росте сложности. Динамический сценарий с изменением матриц переходов каждые 10 итераций проверил устойчивость методов к вариациям среды.

В базовом сценарии с пятью состояниями PI продемонстрировал

сходимость за 2 итерации (14 мс/) с полнотой моделирования 100%, определив оптимальный путь и ожидаемое вознаграждение . VI, напротив, потребовал 5 итераций (32 мс), достигнув полноты 85% при аналогичном пути, но без явного учёта действия (R), что потребовало дополнительных вычислений. Таким образом, PI показал преимущество по скорости сходимости в 2.3 раза и более высокую точность в компактных моделях. При масштабировании числа состояний производительность методов изменилась. Для 20 состояний PI завершил вычисления за 45 мс, а VI — за 78 мс, при этом использование памяти составило 48 МБ и 32 МБ соответственно. При 50 состояниях разрыв в скорости сократился (210 мс для PI против 320 мс для VI), а при 100 состояниях VI стал быстрее (820 мс против 950 мс для PI), сохраняя экономию памяти на уровне 30%. Это объясняется различиями в вычислительной сложности: на итерацию для VI против для PI, где (n) — число состояний, а (m) — число действий. Полнота моделирования также варьировалась в зависимости от масштаба. При 5 состояниях PI обеспечивал 100% против 85% у VI, что обусловлено явным учётом действия (R) и соответствием требованиям MITRE ATLAS. Для 20 состояний преимущество PI составило 28% (98% против 70%), для 50 — 35% (95% против 60%), но при 100 состояниях разрыв сократился до 10% (65% против 55%). VI теряет точность из-за неявного представления политики, тогда как PI сохраняет прозрачность стратегии на каждой итерации. Сравнение характеристик методов выявило их ключевые различия (табл. 1).

Таблица 1- Сравнения по критериям

Критерий	Value Iteration	Policy Iteration
Скорость сходимости	Медленнее (5 итераций в эксперименте)	Быстрее (2 итерации в эксперименте)
Вычислительная сложность	Дешевле на итерацию (обновление значений)	Дороже на итерацию (решение систем уравнений)
Точность результатов	Высокая (совпадает с PI)	Высокая (совпадает с VI)
Распознаваемость стратегии	Неявная (стратегия извлекается из значений в конце вычислений)	Явная (стратегия хранится и обновляется)
Масштабируемость	Лучше для больших пространств состояний	Хуже для больших пространств состояний
Память	Требует хранения значений состояний	Требует хранения политики и значений
Устойчивость к изменениям	Устойчив (динамическое обновление значений)	Менее устойчив (требует пересчета политики)

Метод VI имеет высокую степень масштабируемости (применим к различным множествам состояний разного масштаба) и показывает

устойчивость к изменениям среды благодаря динамическому обновлению значений, требуя при этом меньший объём памяти. Метод PI, показывает более быструю сходимость и распознаваемость (последовательности действий и состояний сразу видны) за счёт хранения политики (, что делает его предпочтительным для задач с явными стратегиями и ограниченным числом действий и состояний. Сходства фиксируются в таблице 2.

Таблица 2 – Сходство методов

Аспект	Value Iteration	Policy Iteration	Сходства
Основная идея	Прямая оптимизация value-функции	Чередование оценки и улучшения политики	Оба гарантируют сходимость к оптимальному решению
Стратегия	Неявная (извлекается в конце)	Явная (хранится на каждом шаге)	Итоговая стратегия идентична
Использование ресурсов	Меньше памяти на итерацию	Больше памяти (хранение политики)	Зависит от размера задачи
Применение	Для задач с большим числом состояний	Для задач с явными стратегиями	Подходят для МППР-моделей

Таким образом, метод PI эффективен в компактных моделях (до 50 состояний). Он обеспечивает быструю сходимость. Метод VI, напротив, выигрывает в крупных системах (более 100 состояний и действий) и динамических условиях, где важны экономия ресурсов и адаптивность. На основе анализа предлагается гибридный подход для оптимизации моделирования угроз. На первом этапе применяется метод PI в случае, если описание системы осложнено большим множеством действий и состояний. Если число состояний и действий меньше 50 применяется метод PI для формирования базовой стратегии с высокой полнотой. На втором этапе метод VI применяется для масштабирования модели в условиях роста числа состояний, минимизируя затраты памяти. Завершающий этап с PI позволяет верифицировать результаты и обеспечить прозрачность документации. Такой алгоритм сокращает время анализа на 20–40% при сохранении полноты на уровне 90–95%.

Проведённое исследование выявило различия в эффективности Value Iteration и Policy Iteration при моделировании кибератак на ИИ-системы в рамках МППР. Метод PI превосходит VI по скорости сходимости (в 2.3 раза) и полноте моделирования (до 35% преимущества) в моделях с малым числом состояний, что делает его оптимальным для задач аудита безопасности. Метод VI, в свою очередь, демонстрирует преимущества в масштабируемости и экономии ресурсов (на 30%) при обработке крупных систем. Предложенный гибридный подход объединяет эти качества,

обеспечивая сбалансированное решение для анализа угроз. Перспективы дальнейших исследований связаны с адаптацией методов к частично наблюдаемым МППР (POMDP) и интеграцией с алгоритмами глубокого обучения для защиты ИИ от состязательных атак, что может расширить их применимость в реальных условиях кибербезопасности.

Список использованных источников

1. ГОСТ Р 59277—2020. Системы искусственного интеллекта. Классификация систем искусственного интеллекта. Утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 23 декабря 2020 г. № 1372-ст. [Электронный ресурс]. – Режим доступа к рес.: https://docs.cntd.ru/document/1200177292 (дата обращения: 30.10.2021).

2. Намиот, Д. Е. Схемы атак на модели машинного обучения / Д. Е. Намиот // International journal of open information technologies. – 2023 – Т.11, № 5. – С. 68-86

3. Бордак, И. В. Разработка метода количественной оценки и прогнозирования безопасности информации ограниченного доступа на основе Марковских случайных процессов / И. В. Бордак, А. П. Росенко // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2017. – Т. 20, № 4. – С. 67–70.

4. Sood, A. K., & Enbody, R. J. A Framework for Modeling Cyber Attacks Using Markov Decision Processes/ A. K. Sood, R. J. Enbody, (2013) [Электронный ресурс]. – Режим доступа: https://arxiv.org/pdf/2207.05436 (дата обращения: 21.03.2021).

5. Кохендерфер, М. Алгоритмы принятия решений / М. Кохендерфер, Т. Уилер, К. Рэй. – Москва: ДМК Пресс, 2023. – 684 с.

6. Саттон, Р. С. Обучение с подкреплением: введение / Р. С. Саттон, Э. Дж. Барто; 2-е изд. – Москва: ДМК Пресс, 2020. – 552 с.

7. MITRE ATLAS // MITRE ATT&CK [Электронный ресурс] – Режим доступа к рес.: https://atlas.mitre.org, свободный (дата обращения: 02.05.2024)

8. Bolum, Wang.Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks/ Bolum Wang, Yuanshum Yao, Shawn Shan, Huiying Li // Conference: 2019 IEEE Symposium on Security and Privacy.–2019.

9. Hu, Z., Beuran, R., & Tan, Y.Automated Penetration Testing Using Reinforcement Learning / Z.Hu, R. Beuran, Y. Tan // 2020. [Электронный ресурс] – Режим доступа к рес.: https://www.researchgate.net/publication/353941853_Using_Cyber_Terrain_in_Reinforcement_Learning_for_Penetration_Testing (дата обращения: 11.12.2023)

Подтопельный Владислав Владимирович, ст. преп. Института цифровых технологий (ИЦТ) Калининградского государственного технического университета (КГТУ), ionpvv@mail.ru.