

А.Е. Самотуга, П.С. Ложников

**КОМПЬЮТЕРНЫЕ АТАКИ НА ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ**

Одной из причин, которая может вызвать снижение доверия, а, значит, и интереса среди руководителей и лиц, принимающих решения в организациях, к внедрению систем на основе искусственного интеллекта является подверженность ИИ разного рода компьютерным атакам. Среди которых особое внимание следует уделить специфическим атакам именно для данной области знаний, использующие знания об уязвимостях ИИ, недостатков выбранных моделей, нехватки обучающих данных и рядом других.

Важным компонентом доверия к ИИ как известно являются его объяснимость, этичность, защищенность. Далее кратко рассмотрим список известных атак на искусственный интеллект, который должен быть принят во внимание при разработке защищенного ИИ.

## Классификация компьютерных атак на ИИ

Выделение классов атак необходимо для организации системного решения проблемы создания ИИ в защищенном исполнении.

Выделяют следующие характерные атаки на ИИ:

* Извлечение данных;
* Извлечение параметров модели для построения алгоритма ее работы;
* Извлечение и интерпретация знаний ИИ;
* Манипуляции входными и обучающими данными;
* Состязательные атаки;
* Зондирование моделей;
* Отказ в обслуживании ИИ.

Перечисленные атаки относятся к двум разновидностям: атаки-манипуляции (направлены на изменение поведения системы путем модификации входных данных, обучающих данных или самой модели) и атаки-извлечения (незаметное получение из системы конфиденциальной информации).

По направлению вектора атаки выделяют:

- Анализ операций, совершаемых ИИ (алгоритма работы ИИ, суть преобразований);

- Управление ИИ (с помощью изменения алгоритма работы, подмены данных ИИ и т.д.);

- Извлечения знаний ИИ.

## Извлечение данных (входных, выходных). Извлечение параметров модели для понимания алгоритма ее работы. Извлечение и интерпретация знаний ИИ

Атаки извлечения входных данных актуальны в ситуации, когда результаты работы модели общедоступны, а входные данные хранятся в секрете.

Так, атаки извлечения обучающих данных (одна из атак инверсии модели) нацелены на получение сведений о наборе данных, применяемых при обучении модели.

Для непрозрачных систем машинного обучения опасность представляют атаки извлечения модели. Они позволяют скопировать поведение или параметры.

В вычислительной части ИИ могут обрабатываться в зависимости от области применения персональные и другие конфиденциальные данные. Возникает задача по защите этих данных от угроз конфиденциальности. Для чего принято заниматься шифрованием параметров решающих правил с применением некоторого криптографического ключа.

Тем не менее есть ряд атак, которые могут обойти данный подход к защите от извлечения данных. Среди них стоит выделить: Атаки «на решающий бит», атаки «ключ под ковриком», «Состязательные» атаки.

Атаки «на решающий бит» (атаки «одного бита»). Есть два пути реализации данной атаки. Первый связан с редактированием программного кода ИИ (точнее его скомпилированной и обученной версии). Если на выходе ИИ возникают короткие команды, то злоумышленник может инвертировать логику программы, изменив решающее правило. Например, если на выходе нейронной сети располагается функция SoftMax, то достаточно поменять два ее выхода местами, чтобы заменить одно управляющее воздействие на другое (например, красный сигнал светофора на зеленый). Второй случай возникает, если хакер напрямую подключится к объекту управления или к каналу передачи данных с возможностью изменять сигналы на выходе ИИ. В этом случае он сможет имитировать определенные управляющие воздействия и изменять одну команду на другую. При этом ему не потребуется вникать в суть работы алгоритма анализа данных, достаточно лишь выявить ассоциации кода команды и связанного с ней действия. Для коротких управляющих команд выявить эти ассоциации несложно.

Другая неприятность связана с тем, что ключ (или ключи) шифрования параметров решающих правил ИИ (например, таблицы связей и весовых коэффициентов искусственных нейронных сетей) должны где-то храниться. Чтобы ИИ «заработал», требуется сначала дешифровать его «мозг». Таким образом, когда алгоритм анализа входных данных выполняется приложением, параметры решающих правил остаются незащищенными. В теории гомоморфное шифрование может быть использовано для защиты ИИ, но на практике до этого еще далеко, так как имеются нерешенные проблемы с низкой производительностью и с накоплением ошибок. Если не использовать гомоморфное шифрование, то ключи должны где-то храниться (в базе данных, в коде ИИ и т.д.), что создает ряд внешних и внутренних угроз. Злоумышленник может похитить ключ, используя уязвимости в защите или вступив в сговор администратором. Человек всегда является «узким местом» в системе безопасности, поэтому эти вопросы нельзя закрыть полностью. Хуже всего, если все знания ИИ шифруются на одном ключе (нет разделения знаний на независимо зашифрованные фрагменты).

«Состязательные» атаки (спуфинг), при которых хакер подает на вход ИИ фальсифицированные или перехваченные данные с целью получения на выходе ИИ желаемых управляющих воздействий. К данной категории также относится атака «извлечения знаний» из нейронной сети. Под этим термином подразумевается частичное или полное восстановление обучающей выборки путем наблюдения статистики входов/выходов нейронной сети во время ее работы, либо путем непосредственного анализа параметров обученной нейронной сети в незашифрованном виде (например, таблиц связей и весовых коэффициентов нейронов). Конфиденциальная информация и персональные данные, находящиеся в памяти нейронной сети, не должны быть извлечены злоумышленником, даже если ее параметры хранятся в незашифрованном виде. Отметим, что реализация такой концепции как федеративное обучение (заключение модели ИИ в защищенную среду и ее обучение без перемещения обучающей выборки куда-либо) не дает защиты от атаки «извлечения знаний», так как эта атака направлена на параметры уже обученного ИИ, при условии, что процесс обучения мог уже проходить в защищенной среде. Отметим, что подобная атака может осуществляться в отношении не только нейросетевых реализаций ИИ.

1. **Манипуляции с моделями. Манипуляции входными и обучающими данными. Состязательные атаки. Зондирование моделей**

Манипуляции с обучающими данными (атаки отравления) предполагает, что атакующий создает входные данные для действующей системы, и при обработке выдается не тот результат, на который рассчитывали создатели системы: неверная классификация знака остановки, ошибочная идентификация спама, неверная интерпретация высказываний.

Манипуляции с обучающими данными позволяют воздействовать на рабочую модель в процессе обучения. Злоумышленник модифицирует корпус данных, применяемый для обучения системы, с целью влияния на ее дальнейшее поведение. Например, атакующий может опубликовать неверные данные для искажения процессов медицинской диагностики или финансового прогнозирования.

Атаки путем манипуляции с моделями менее известны и распространены. Типовой сценарий данной атаки выглядит следующим образом: атакующий публикует модель — «прозрачный ящик», в которую заложено некоторое скрытое поведение с расчетом на то, что испорченной моделью воспользуются третьи стороны, что особенно актуально с учетом роста популярности применения предобученных моделей.

Приведем несколько примеров атак, связанных с манипуляциями данных злоумышленников. Во-первых, искажение разметки, чтобы провести такую атаку на алгоритм машинного обучения, злоумышленник должен получить доступ к обучающей выборке и добавить в выборку объекты с неправильной разметкой. Обученная на искаженной выборке, ML-модель будет ошибаться и на других схожих объектах. Второе направление - искажение обучающей выборки. Имея доступ к обучающей выборке, злоумышленник может добавить в неё специальные объекты, которые ухудшают качество работы прогностической модели. При этом разметка может быть корректной, однако сами объекты являются нестандартными. Например, это может быть файл, который по каким-то параметрам значительно отличается от типичных файлов для данной задачи («чёрный лебедь»).

Это угроза особенно серьёзна в связи с тем, что многие разработчики ML-решений, включая производителей систем безопасности «следующего поколения», используют публичные обучающие выборки, которые могут быть легко «отравлены» третьими сторонами.

Это угроза особенно серьёзна в связи с тем, что многие разработчики ML-решений, включая производителей систем безопасности «следующего поколения», используют публичные обучающие выборки, которые могут быть легко «отравлены» третьими сторонами.

Если исходный код недоступен, злоумышленник может атаковать ML-модель «грубым перебором», то есть делать небольшие изменения во вредоносных файлах и раз за разом тестировать модель на этих изменённых файлах – до тех пор, пока не будет обнаружено её слабое место. Подобная атака «чёрного ящика» является довольно трудоёмкой, однако атакующий может её автоматизировать, используя «враждебный» искусственный интеллект (adversarial AI) для генерации большого количества вредоносных образцов.

Также следуют упомянуть момент, связанный с предуобученными моделями и аутсорсингом ML-моделей. Недостаток ресурсов зачастую вынуждает разработчиков использовать чужие архитектуры машинного обучения, созданные сторонними разработчиками для решения стандартных задач обработки данных. Эти готовые и всем доступные решения могут быть хорошо знакомы злоумышленникам, что облегчает проведение атаки «белого ящика». Ещё один вектор атаки связан с аутсорсингом: некоторые ML-модели обучаются сторонними командами экспертов или публичными ML-сервисами. Такие модели могут содержать бэкдоры.

Злоумышленник, не имеющий доступа к обучающей выборке, всё равно может активно взаимодействовать с системой машинного обучения. В частности, если атакующий получил доступ к самой модели в локальном пользовательском продукте, он может скрытно исследовать модель, сколько пожелает, и даже воссоздать её исходный код. Так он узнает архитектуру модели и выявит, какие признаки объектов используются. После этого злоумышленник может создать вредоносное ПО, которое будет маскировать или усиливать нужные признаки. Это называют атакой «белого ящика» или «кражей модели»

1. **Отказ в обслуживании ИИ**

Чтобы оценить надежность системы ИИ следует в первую очередь традиционные требования безопасности ИТ, к примеру в ИСО/МЭК 27001, ИСО/МЭК 18045 и ИСО/МЭК 62443. Эти стандарты обеспечивают процессы аудита и сертификации требований безопасности ИТ, которые также применимы к системам ИИ. помимо следования передовым методам и соблюдения существующих стандартов для обычных систем, ИИ сопряжен с целым рядом проблем, которые необходимо учитывать при обсуждении надежности, особенно в контексте функциональной безопасности. Модели систем ИИ, особенно более сложные (например, нейронные сети), могут иметь определенные недостатки, которых нет в других типах систем, и поэтому требуют дополнительной проверки при развертывании в контексте функциональной безопасности.

Состязательное машинное обучение может использоваться для организации сбоя модели ИИ. Можно заставить модель выводить совершенно разные результаты, добавляя к входным данным незначительные возмущения. Этот шум в случае входных изображений, как правило, незаметен для людей, а также может быть столь же хорошо скрыт в числовых входных данных. Хотя эти возмущения, как правило, неслучайны и тщательно обрабатываются с помощью средств процесса оптимизации, нельзя исключать, что аппаратные сбои или системный шум, уже присутствующие во входных данных, вызывают заметный сдвиг в выходных данных модели. Существует возможность физических атак на модели, например можно ввести состязательные примеры в процесс прямого вывода модели, создавая возмущения с помощью физических наклеек, нанесенных на объекты, и вызывая сильно расходящиеся результаты классификации.

Сбои в оборудовании могут нарушать правильное выполнение алгоритма, нарушая его поток управления, вызывая ошибки в памяти, мешая входным данным (таким как сигналы датчиков) и, как правило, вызывают ошибочные результаты или прямое нарушение результатов из-за поврежденных выходов. Надежное оборудование в системах ИИ так же необходимо, как и в обычных системах.

Оборудование, используемое для выполнения технологии ИИ, также может страдать от случайного отказа оборудования. Список соответствующих моделей отказов можно найти в таких стандартах, как IEC 61508-2 и ISO 26262 11.

1. **Методы противодействия атакам на ИИ. Программные продукты для детектирования и отражения атак на ИИ**

Предотвратить атаки манипуляции с входными данными поможет знание внутреннего представления информации в системе для всего возможного пространства входных данных, а не только обучающего распределения. Представления, характеризующиеся нестабильностью, уязвимостью для коррекции, могут быть легко и незаметно искажены, поэтому оптимизация принципов представления данных позволяет улучшить безопасность систем машинного обучения. Также система будет защищена от отказов в ситуациях, когда данных недостаточно или они слишком зашумлены.

Другое направления защиты от манипуляций с входными данными представляет собой методы распознавания аномалий. Такой подход предполагает выбор и исследование какой-либо меры типичности данных, что позволяет защититься от отравления модели.

Для устранения угрозы от атаки искажения разметки следует применять двойную проверку и использование ансамблей методов.

С целью защиты от искажения обучающей выборки, помимо использования обнаружения аномалий, также рекомендуется содержать выборки в тайне. Внедрение оценки уровней доверия для прогноза и использование многоуровневой защиты также является подходящим направлением для решения данной проблемы.

При разработке защиты атаки “черного” и “белого” ящика следует выполнять машинное обучение в облаке, исследовать модель на доказуемую устойчивость.

Для нивелирования возможности атаки на предобученные и аутсорсинговые модели, простейшим решением является самостоятельная разработка моделей, в идеале на собственной аппаратной базе.

При возможной утечке через обученные модели подходящее решение - грамотное ограничение доступа к самой модели и анонимизация данных.

Наконец, важно разрабатывать методы, независимые от локальной архитектуры, что возможно, если отказаться от запуска ML моделей на конечных устройствах.

Некоторые конкретные проблемы, связанные со случайным отказом оборудования ИИ, включают:

— По сравнению с обычным программным обеспечением вычисления с использованием технологии ИИ используют гораздо большие объемы данных и более сложные арифметические вычисления, в зависимости от типов используемых моделей. Это может привести к более высокой вероятности того, что неисправности станут фактическими отказами, т. е. к более высокой вероятности отказа по запросу в час. Также предполагается, что некоторая внутренняя избыточность вычислений, встроенная в модели ИИ, может в некоторой степени подавлять влияние мягких ошибок.

— Обучение модели, с использованием методов машинного обучения, и ее последующее выполнение обычно происходят в разных системах. Поскольку сбои, возникающие на этапе обучения и в работе системы ИИ, могут повлиять на правильное выполнение алгоритма то, как система, используемая для обучения, так и система, используемая для выполнения технологии ИИ, важны для обеспечения функциональной безопасности.

— Облачные, встроенные и графические системы могут иметь особые режимы отказа, которые необходимо устранять.

В числе прочего выделяют отказ по общей причине ( различий между обычным оборудованием и оборудованием на основе ИИ нет, см. IEC 61508-2 и ISO 26262-11) и систематический сбой.

По мере расширения спектра систем искусственного интеллекта встроенные системы приобретают все большее значение. На этапе обучения объем данных и вычислительная мощность, необходимые для расчета параметров модели с помощью алгоритма машинного обучения, могут быть очень высокими, что обычно исключает использование встроенных систем. Когда фаза обучения завершена, рассчитанные параметры передаются в целевую систему. Эта асимметрия методов машинного обучения означает, что на этапе вывода требуется гораздо меньшая вычислительная мощность, и поэтому встроенные системы могут подходить для выполнения системы ИИ.

Следует отметить, что во встроенных системах могут возникнуть трудности с реализацией результатов обучения. Ошибки трансляции могут возникать, например, из-за несовместимости с постоянным запоминающим устройством (ПЗУ) и оперативным запоминающим устройством (ОЗУ) управления микроконтроллером (MCU), микропроцессором (MPU) или цифровым сигнальным процессором (DSP).

При построении систем крайне важно использовать технологии, доказавшие свою технологическую зрелость. Технологическая зрелость описывает, насколько зрелой и безошибочной является конкретная технология в конкретном контексте приложения. Если при разработке системы ИИ используются менее зрелые и новые технологии, они могут создавать риски для безопасности, которые неизвестны или трудно поддаются оценке. Для зрелых технологий обычно доступно большее разнообразие данных об опыте, что упрощает идентификацию и оценку рисков безопасности. Однако зрелые технологии сопряжены с опасностью снижения осведомленности об их потенциальном влиянии на риски для безопасности с течением времени, так что положительный эффект зависит от постоянного мониторинга рисков для безопасности, а также надлежащего обучения и технического обслуживания.

**Список источников**

1. AI в защищенном исполнении. Что это означает и зачем нужны преобразователи образов в код? // VK [Электронный ресурс]. URL: <https://vk.com/@a.i.constructor-iskusstvennyi-intellekt-v-zaschischennom-ispolnenii-chto-eto> (дата обращения: 22.04.2022).

2. Атаки на искусственный интеллект // VK [Электронный ресурс]. URL: <https://media.kaspersky.com/ru/business-security/attacks-on-artificial-intelligence-whitepaper.pdf> (дата обращения: 22.04.2022).

3. Обеспечение безопасности систем машинного обучения // Издательство «Открытые системы» [Электронный ресурс]. URL:<https://www.osp.ru/os/2019/04/13055230/> (дата обращения: 22.04.2022).