# Атаки на системы машинного обучения – общие проблемы и методы

Е.А. Ильюшин, Д.Е. Намиот, И.В. Чижов

***Аннотация*—В работе рассматривается проблема атак на системы машинного обучения. Под такими атаками понимаются специальные воздействия на элементы конвейера машинного обучения (тренировочные данные, собственно модель, тестовые данные) с целью либо добиться желаемого поведения системы, либо воспрепятствовать ее корректной работе. В целом, эта проблема является следствием принципиального момента для всех систем машинного обучения – данные на этапе тестирования (эксплуатации) отличаются от таковых же данных, на которых система обучалась. Соответственно, нарушение работы системы машинного обучения возможно и без целенаправленных действий, просто потому, что мы столкнулись на этапе эксплуатации с данными, для которых генерализация, достигнутая на этапе обучения, не работает. Атака на систему машинного обучения – это, фактически, целенаправленное выведение системы в область данных, на которых система не тренировалась. На сегодняшний день, эта проблема, которая, в общем случае, связана с устойчивостью работы систем машинного обучения, является главным препятствием для использования машинного обучения в критических приложениях.**

***Keywords*—состязательные атаки, кибербезопасность систем машинного обучения**

## I. Введение

Эта статья представляет собой расширенное изложение доклада на конференции MSU AI [1]. Статья является продолжением серии публикаций, посвященных устойчивым моделям машинного обучения [2], [3], [4]. Она подготовлена в рамках проекта кафедры Информационной безопасности факультета ВМК МГУ имени М.В. Ломоносова по созданию и развитию магистерской программы «Искусственный интеллект в кибербезопасности» [5].

Машинное обучение стало важным компонентом многих IT-систем. На сегодняшний день машинное обучение является практическим синонимом термина Искусственный Интеллект, программы развития которого являются уже национальными программами во многих странах. Добавлять в приложения возможности машинного обучения становится все проще. Многие библиотеки машинного обучения и онлайн-сервисы уже не требуют глубоких знаний в области машинного обучения.

Однако даже у простых в использовании систем машинного обучения есть свои проблемы. Среди них - угроза состязательных атак, которая стала одной из важных проблем приложений машинного обучения. Под этим понимается специальное воздействие на элементы конвейера (пайплайна) системы машинного обучения (то есть, воздействию могут подвергаться тренировочные данные, тестовые данные или даже сами модели), призванные вызвать желаемое поведение работающей системы. Таковым поведением может быть, например, неверная работа классификатора. Но существуют и атаки, которые направлены на выяснение параметров работы системы. Эта информация поможет атакующему создать обманывающие систему примеры. Существуют атаки, которые позволяют проверить, например, принадлежность определенных данных к тренировочному набору и, возможно, раскрыть тем самым конфиденциальную информацию.

Воздействие здесь – это специальный подбор тренировочных или тестовых данных, скрытая модификация модели, специальная организация опроса системы и т.д.

## II. Атаки на системы машинного обучения

Термин «состязательные» (или «опровергающие») и нужно понимать в смысле противодействия работе системы машинного обучения (нейронной сети) [4].

Состязательные атаки отличаются от других типов угроз безопасности (атак на ИТ-системы). Поэтому первым шагом к противодействию им является их классификация, понимание их типов, равно как и мест приложения (того, где и атакуются системы машинного обучения).

Природа атак на системы машинного обучения и глубокого обучения отличается от других киберугроз. Состязательные атаки опираются на сложность глубоких нейронных сетей и их статистический характер, чтобы найти способы их использования и изменения их поведения. Нет возможности обнаружить злоумышленные действия с помощью классических инструментов, используемых для защиты программного обеспечения от киберугроз.

Состязательные атаки манипулируют поведением моделей машинного обучения. Большая часть примеров относится к работе с изображениями, но в реальности есть примеры атак на системы анализа текстов, классификации аудио-данных (распознавание речи), анализа временных рядов. В целом, их можно рассматривать как некоторый универсальный риск для

Статья получена 12 января 2022.

Е.А. Ильюшин – МГУ им. М.В. Ломоносова, (email: eugene@ilyushin.science)

Д.Е. Намиот – МГУ им. М.В. Ломоносова, (email: dnamiot@gmail.com)

И.В. Чижов – МГУ им. М.В. Ломоносова, (email: ichizhov@cs.msu.ru)

Таблица I
Классификация атак.

| Атака | Этап | Затрагиваемые параметры |
|---|---|---|
| Adversarial attack | применение | входные данные |
| Backdoor attack | тренировка | параметры сети |
| Data poisoning | тренировка, использование | входные данные |
| IP stealing | использование | отклик системы |
| Neural-level trojan | тренировка | отклик системы |
| Hardware trojan | аппаратное проектирование | отклик системы |
| Side-channel attack | использование | отклик системы |

моделей машинного обучения (глубокого обучения) [6].

Есть разные попытки объяснить природу их существования. По одним гипотезам, состязательные атаки существуют из-за нелинейного характера систем, что ведет к существованию некоторых неохваченных алгоритмом генерализации областей данных. По другим – это наоборот переобучение системы, когда даже небольшие отклонения от тренировочного набора данных обрабатываются неверно.

Термин состязательная атака часто используется в широком смысле для обозначения различных типов злонамеренных действий против моделей машинного обучения. Но состязательные атаки различаются в зависимости от того, на какую часть конвейера машинного обучения они нацелены, от эффекта, которого они добиваются, и от знания об атакуемой системе (черный и белый ящик). В таблице I представлен пример классификации атак.

Теперь формализуем понятие «состязательная атака», для этого необходимо дать формальное определение модели угроз на примере задачи классификации, которая заключается в том, что имея исходный набор данных $X$ и конечное множество меток классов $Y$ необходимо найти отображение $f : X \to Y$. Тогда отображение $f$ уязвимо для состязательных атак тогда, когда существует отображение $A$ такое, что для любого $x \in X$ найдется $\tilde{x} = A(x)$, для которого $f(\tilde{x}) \neq y$ при том, что $f(x) = y$.

Теперь мы можем формально определить состязательную атаку следующим образом:

*пусть $x \in R^d$ принадлежит классу $y$, тогда состязательная атака – это отображение $A : R^d \to R^d$ такое что $\tilde{x} = A(x)$ и $f(\tilde{x}) = y_t$.*

Среди всевозможных видов состязательных атак можно выделить подмножество аддитивных состязательных атак, которые определяются так:

*пусть $x \in R^d$ принадлежит классу $y$, тогда аддитивная состязательная атака – это добавление некоторого шума $\eta \in R^d$ к $x$, так что $\tilde{x} = x + \eta$ и $f(\tilde{x}) = y_t$.*

Стоит отметить ряд важных свойств аддитивных атак, а именно то, что такие атаки гарантируют неизменность входного пространства и они интерпретируемы. Большинство существующих состязательных атак именно этого типа. Аддитивные атаки в свою очередь можно разделить, как минимум, на три типа. Для того что бы перейти к формализации указанных типов, давайте рассмотрим задачу нарушения работы классификатора в следующем виде. Дано множество классов $\{y_1, y_2, y_3, \ldots, y_k\}$. Границы классов заданы дискриминирующими функциями $\{g_1(\cdot), g_2(\cdot), \ldots, g_k(\cdot)\}$. Задача состоит в нарушении работы классификатора так, чтобы $f$ сопоставлял $x$ класс $y_t$, для этого необходимо чтобы:

$$g_t(\tilde{x}) \geq g_i(\tilde{x}), \text{ для всех } i \neq t \qquad \text{(II.1)}$$

Тогда значение $g_t(x)$ должно быть большем, чем значение любой другой $g(\cdot)$ или:

$$g_t(\tilde{x}) \geq \max_{i \neq t}\{g_i(\tilde{x})\} \iff \max_{i \neq t}\{g_i(\tilde{x})\} - g_t(\tilde{x}) \leq 0 \qquad \text{(II.2)}$$

Неравенство II.2 накладывает ограничение на множество атакующих $\tilde{x}$. Для того чтобы величина искажения $\eta$ было как можно меньше, $\tilde{\tilde{x}}$ должен быть как можно ближе к $x$. Раз у нас появился такой критерий, то мы можем формализовать три типа аддитивных атак:

1) *Атака минимизирующая функция расстояния*, где $\|\cdot\|$ любая функция расстояния:

$$\min_x \|\tilde{x} - x\|, \text{ при условии } \max_{i \neq t}\{g_i(\tilde{x})\} - g_t(\tilde{x}) \leq 0 \qquad \text{(II.3)}$$

2) *Атака максимизирующая функцию расстояния*, где $\|\cdot\|$ любая функция расстояния, а $\lambda > 0$ – задает верхнюю границу для атаки:

$$\min_x \max_{i \neq t}\{g_i(\tilde{x})\} - g_t(\tilde{x}), \text{ при условии } \|\tilde{x} - x\| \leq \lambda \qquad \text{(II.4)}$$

3) *Атака на основе регуляризации*, где $\|\cdot\|$ любая функция расстояния, а $\alpha > 0$ – параметр регуляризации:

$$\min_x \|\tilde{x} - x\| + \alpha(\max_{i \neq t}\{g_i(\tilde{x})\} - g_t(\tilde{x})) \qquad \text{(II.5)}$$

Основной проблемой в части борьбы с атаками является отсутствие универсальных методов их предотвращения. Все развитие в этой области до сих пор следует бесконечному циклу: появление новой атаки – модификация алгоритма и/или модели для ее предотвращения (смягчения последствий) – появление новой атаки преодолевающей защиту и т.д.

Это были примеры атак, которые воздействуют на входные данные. Другой пример – атаки, которые пытаются воздействовать на тренировочные данные. Отравление – это как раз пример таких атак. Отравление может быть разным, как и уклонение. Такие атаки также могут быть целевыми и нецелевыми. Далее такие атаки могут различаться по допустимым (возможным) воздействиям на тренировочные данные. Например, можно ли добавлять новые размеченные данные или только менять метки у существующих данных и т.п.

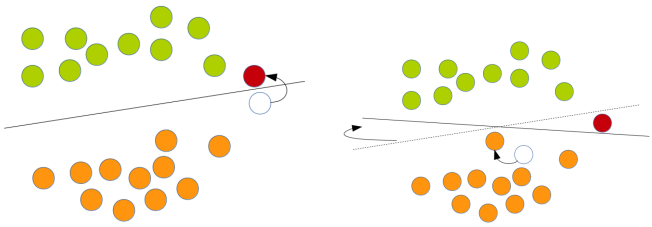
В [7] отмечено четыре общих стратегии атак отравлением:

- *Модификация меток*: эти атаки позволяют злоумышленнику изменять только метки в наборах данных контролируемого обучения, но для произвольных точек данных. Как правило, с учетом ограничения на общую стоимость модификации.
- *Внедрение данных*: злоумышленник не имеет никакого доступа к обучающим данным, а также к алгоритму обучения, но имеет возможность добавлять новые данные в обучающий набор. Можно

испортить целевую модель, вставив состязательные образцы в набор обучающих данных.

- *Модификация данных*: злоумышленник не имеет доступа к алгоритму обучения, но имеет полный доступ к обучающим данным. Обучающие данные могут быть отравлены напрямую путем изменения данных перед их использованием для обучения целевой модели.
- *Логическое искажение*: злоумышленник имеет возможность вмешиваться в алгоритм обучения. Эти атаки называются повреждением логики.

На рисунке 1 проиллюстрирована атака отравлением на классификатор на базе SVM. Отравление – это не изменение входных данных, а изменение самой гиперплоскости классификатора. Отравляющие атаки меняют границу классификации, а состязательные атаки меняют входной пример.




| | Adversarial example | Data poisoning |
|---|---|---|
| Pros | simple way to bypass a defense | allows more types of attacks |
| Cons | requires owning the testing data | requires owning the training data |

Рис. 1. Сравнение атак [8].

Есть также два других типа атак, таких как бэкдоры и трояны, технически они очень похожи на атаки отравления. Разница заключается в том, какие данные доступны атакующему. Бэкдор-атака внедряет бэкдор в модель машинного обучения таким образом, что модель с бэкдором учиться решать как выбранную злоумышленником подзадачу, так и основную задачу [9]. С одной стороны, модель с бэкдором ведет себя нормально, как ее исходная модель-аналог на входных данных, не содержащих триггера, что делает невозможным отличить модель с бэкдором от исходной модели, проверяя точность модели только с помощью тестовых образцов. Это отличается от вышеупомянутой атаки отравления, которая ухудшает общую точность основной задачи, поэтому становится заметной или подозрительной. С другой стороны, модель с бэкдором выполняет задачу злоумышленника после того, как секретный триггер добавлен во входные данные, независимо от исходного содержимого входных данных.

Формально бэкдор атаку можно определить следующим образом: есть исходный $x_i$, на котором предсказанная метка $z_a = F_{\theta_{bd}}(x_i)$ модели с бэкдором имеет с высокой вероятность туже метку что и предсказание, полученнное на модели без бэкдора. Но на $x_i^a = x_i + \delta$, где $\delta$ – является триггером, модель с бэкдором будет всегда предсказывать метку $z_a$ необходимую злоумышленнику, причем в некоторых случаях не взирая на то, что собой представляет исходный $x_i$. Однако для чистых входных данных модель с бэкдором ведет себя как исходная модель без (ощутимого) ухудшения производительности. Отметим, что большинство атак и контрмер через бэкдор сосредоточены на триггере, не зависящем от ввода или не зависящем от класса. Однако в некоторых исследованиях основное внимание уделяется триггеру, зависящему от класса, или триггеру, зависящему от класса.

Успех бэкдор атаки обычно можно оценить по точности чистых данных (CDA – Clean Data Accuracy) и коэффициенту успешности атаки (ASR – attack success rate), которые можно определить следующим образом[10]:

- *Clean Data Accuracy (CDA)* – это доля чистых тестовых примеров, не содержащих триггера, для которых правильно предсказаны метки классов.
- *Attack Success Rate (ASR)* – - это доля чистых тестовых примеров с триггером, которые классифицируются согласно целям злоумышленника.

Для успешной модели с бэкдорами CDA должен быть равен CDA модели без бэкдоров, а ASR должен стремиться к 100%, такого высокого показателя ASR удается как правило достичь при аутсорсинге модели (обучение на сторонних вычислительных ресурсах).

Поверхности бэкдор атак:

- *Отравление кода* – специалисты по машинному обучению часто используют такие фреймворки как Caffe, TensorFlow и Torch для ускорения разработки моделей машинного обучения. Эти фреймворки часто создаются на основе программных пакетов сторонних производителей, которые могут быть нетривиальными для аудита или тестирования. Следовательно, использование общедоступных фреймворков DL может привести к появлению уязвимостей в моделях машинного обучения, построенных с помощью них. Злоумышленник может использовать уязвимости для выполнения различных атак, начиная от атак типа «отказ в обслуживании» приложения DL и заканчивая атаками с перехватом потока управления, которые либо скомпрометируют систему, либо ускользнут от обнаружения [11].
- *Аутсорсинг* – пользователь передаст обучение модели третьей стороне из-за отсутствия у него навыков машинного обучения или вычислительных ресурсов. В этом сценарии пользователь определяет архитектуру модели, предоставляет данные для обучения и передает обучение машинному обучению как услуге (MLaaS). Таким образом, злонамеренный провайдер MLaaS контролирует фазу обучения и интегрирует бэкдоры в модели в процессе обучения.
- *Предобученная обученная* – эта поверхность атаки появляется при повторном использовании предварительно обученной модели или модели «учителя». С одной стороны, злоумышленник может выпустить и рекламировать модель с бэкдорами для извлечения признаков, например, зоопарк моделей, используемый жертвой для трансферного обучения [12]. При обработке естественного языка word embedding мо-

жет действовать как средство извлечения признаков, которым также могли злонамеренно манипулировать [13]. Трансферное обучение – обычное дело для обучения модели «студент» при недостаточном количестве обучающих данных. Обычно это тот случай, когда сбор данных и маркировка требуют больших затрат и специальных знаний. Кроме того, трансферное обучение также снижает накладные расходы на вычисления. С другой стороны, злоумышленник сначала загружает популярную предварительно обученную модель, манипулирует исходной моделью и повторно обучает ее с помощью созданных данных (то есть, внедряя в модель бэкдор). После этого злоумышленник выкладывает модель с бэкдором в открытый доступ [14].

- *Сбор данных* – этап сбора данных обычно является источником ошибок, так-ка часто используются не надежные источники. Если пользователь собирает обучающие данные из нескольких источников, то атаки с отравлением данных становятся более реальной угрозой. Например, существуют популярные и общедоступные наборы данных, которые основаны на вкладе волонтеров или получении данных из Интернета, например, ImageNet. OpenAI обучает модель GPT-2 на всех веб-страницах, где взаимодействовали по крайней мере три пользователя Reddit. Таким образом, некоторые собранные данные могли быть искажены. Такие атаки с отравлением данных сохраняют согласованность между метками и значениями данных, обходя, таким образом, ручной или визуальный контроль.
- *Распределенное обучение* – этот сценарий касается распределенного обучения (федеративное обучение и раздельное обучение) [15]. Например, Google обучает модели предсказания слов на основе данных, локализованных на телефонах пользователей [16]. Распределенное обучение предназначено для защиты от утечки конфедециальных данных, принадлежащих клиентам. На этапе обучения сервер не имеет доступа к данным обучения участников. Это делает распределенное обучение уязвимым для различных атак, включая бэкдор атаку. В обученную модель может быть легко интегрированы бэкдоры, когда очень небольшое количество участников скомпрометировано или контролируется злоумышленником. Как локальное заражение данных, так и заражение модели может быть выполнено злоумышленником, чтобы внедрить бэкдор в модель.
- *В промышленной эксплуатации* – такая бэкдор-атака происходит после развертывания модели машинного обучения, особенно на этапе вывода [17]. Как правило, веса модели изменяются путем fault injection (например, лазером, напряжением и т.д.). Рассмотрим типичный сценарий атаки, когда злоумышленник и пользователь являются двумя процессами, совместно использующими один и тот же сервер. Пользователь запускает модель машинного обучения и загружает веса машинного обучения в память. Злоумышленник косвенно меняет биты весов, вызывая ошибку rowhammer [18], что приводит к снижению точности вывода. Отметим, что такую атаку невозможно предотвратить с помощью оффлайн проверки.

Атаки, направленные на кражу интеллектуальной собственности призваны выявить существенные факты о тренировочных наборах, самой модели или ее параметрах. Например, атаки на вывод (они же – инференс атаки, в оригинале – Membership Inference атаки) – призваны проверить, находится ли конкретный экземпляр данных в обучающем наборе. Базовая идея – модель ведет себя по-разному на входных данных, которые были в обучающем наборе или не были в нем. В классической работе [19] строится ряд теневых моделей (моделей, подобных исходной) на которых оцениваются искусственно созданные наборы данных, включающие интересующие нас примеры.

Атаки инверсией модели (атаки извлечения данных) пытаются фактически извлечь данные из тренировочного набора. Например, при работе с изображениями можно попытаться извлечь определенное изображение из набора данных [20]. Отметим, что такие атаки связаны, естественным образом, с множественными запросами к модели и оценкой ее ответов. Трудно предположить, что модели для критических применений будут иметь какой-то открытый интерфейс, позволяющий сторонние запросы. Так что такие атаки относятся, скорее, к каким-то публичным сервисам (MLaaS – machine learning as service [21]).

Вывод параметров или извлечение модели – еще одна атака, целью которой является восстановление точной модели или даже ее гиперпараметров [22]. Идея о том, как это может быть сделано, представлена на рисунке 2 [23]. Датасет обрабатывается порциями, для того, чтобы понять структуру модели.


Рис. 2. Извлечение модели.

## III. Заключение

Для критических применений остро стоит вопрос сертификации систем, моделей и наборов данных, подобно тому, как это делается для традиционных систем программного обеспечения, используемых в таких приложениях. Важно также, что состязательные атаки вызывают проблемы с доверием к алгоритмам машинного обучения, особенно к глубоким нейронным сетям. Дискуссии об этике систем искусственного интеллекта связаны, в том числе, и с состязательными атаками.

## IV. Благодарности

Мы благодарны сотрудникам кафедры Информационной безопасности факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова за ценные обсуждения данной работы.

## Список литературы


[1] MSU AI. [Online]. Available: https://event.msu.ru/aiconference

[2] D. Namiot, E. Ilyushin, and I. Chizov, “Ongoing academic and industrial projects dedicated to robust machine learning,” vol. 9, no. 10, pp. 35–36.

[3] D. Namiot, E. Ilyushin, and I. Chizov, “Military applications of machine learning,” vol. 10, no. 1, pp. 69–76.

[4] D. Namiot, E. Ilyushin, and I. Chizov, “The rationale for working on robust machine learning,” vol. 9, no. 11, pp. 68–74.

[5] Artificial intelligence in cybersecurity. [Online]. Available: http://master.cmc.msu.ru/?q=ru/node/3496(inRussian)

[6] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning.” [Online]. Available: http://arxiv.org/abs/1712.07107

[7] How to attack machine learning (evasion, poisoning, inference, trojans, backdoors). [Online]. Available: https://bit.ly/3iCBKmf

[8] A. Chan-Hon-Tong, “An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning,” vol. 1, no. 1, pp. 192–204. [Online]. Available: http://www.mdpi.com/2504-4990/1/1/11

[9] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning.” [Online]. Available: http://arxiv.org/abs/1712.05526

[10] A. K. Veldanda, K. Liu, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, B. Dolan-Gavitt, and S. Garg, “NNoculation: Catching BadNets in the wild,” pp. 49–60. [Online]. Available: http://arxiv.org/abs/2002.08313

[11] Q. Xiao, K. Li, D. Zhang, and W. Xu, “Security risks in deep learning implementations,” in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, pp. 123–128. [Online]. Available: https://ieeexplore.ieee.org/document/8424643/

[12] Y. Ji, X. Zhang, S. Ji, X. Luo, and T. Wang, “Model-reuse attacks on deep learning systems.” [Online]. Available: http://arxiv.org/abs/1812.00483

[13] R. Schuster, T. Schuster, Y. Meri, and V. Shmatikov, “Humpty dumpty: Controlling word meanings via corpus poisoning.” [Online]. Available: http://arxiv.org/abs/2001.04935

[14] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, “Trojaning attack on neural networks,” 2017.

[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning.” [Online]. Available: http://arxiv.org/abs/1807.00459

[16] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, “Federated learning for mobile keyboard prediction.” [Online]. Available: http://arxiv.org/abs/1811.03604

[17] A. S. Rakin, Z. He, and D. Fan, “TBT: Targeted neural network attack with bit trojan.” [Online]. Available: http://arxiv.org/abs/1909.05193

[18] S. Hong, P. Frigo, Y. Kaya, C. Giuffrida, and T. Dumitraş, “Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks.” [Online]. Available: http://arxiv.org/abs/1906.01017

[19] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models.” [Online]. Available: http://arxiv.org/abs/1610.05820

[20] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,” in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 739–753. [Online]. Available: https://ieeexplore.ieee.org/document/8835245/

[21] M. Ribeiro, K. Grolinger, and M. A. Capretz, “MLaaS: Machine learning as a service,” in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 896–902. [Online]. Available: http://ieeexplore.ieee.org/document/7424435/

[22] H. Yan, X. Li, H. Li, J. Li, W. Sun, and F. Li, “Monitoring-based differential privacy mechanism against query flooding-based model extraction attack,” pp. 1–1. [Online]. Available: https://ieeexplore.ieee.org/document/9389670/

[23] B. Wang and N. Z. Gong, “Stealing hyperparameters in machine learning.” [Online]. Available: http://arxiv.org/abs/1802.05351

# Attacks on machine learning systems – common problems and methods

Eugene Ilyushin, Dmitry Namiot, Ivan Chizhov

***Abstract*—The paper deals with the problem of adversarial attacks on machine learning systems. Such attacks are understood as special actions on the elements of the machine learning pipeline (training data, the model itself, test data) in order to either achieve the desired behavior of the system or prevent it from working correctly. In general, this problem is a consequence of a fundamental moment for all machine learning systems - the data at the testing (operation) stage differs from the same data on which the system was trained. Accordingly, a violation of the machine learning system is possible without targeted actions, simply because we encountered data at the operational stage for which the generalization achieved at the training stage does not work. An attack on a machine learning system is, in fact, a targeted introduction of the system into the data area on which the system was not trained. Today, this problem, which is generally associated with the stability of machine learning systems, is the main obstacle to the use of machine learning in critical applications.**

***Keywords*—machine learning, cyberattacks, adversarial examples**

## Список литературы

[1] MSU AI. [Online]. Available: https://event.msu.ru/aiconference

[2] D. Namiot, E. Ilyushin, and I. Chizov, "Ongoing academic and industrial projects dedicated to robust machine learning," vol. 9, no. 10, pp. 35–36.

[3] D. Namiot, E. Ilyushin, and I. Chizov, "Military applications of machine learning," vol. 10, no. 1, pp. 69–76.

[4] D. Namiot, E. Ilyushin, and I. Chizov, "The rationale for working on robust machine learning," vol. 9, no. 11, pp. 68–74.

[5] Artificial intelligence in cybersecurity. [Online]. Available: http://master.cmc.msu.ru/?q=ru/node/3496(inRussian)

[6] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning." [Online]. Available: http://arxiv.org/abs/1712.07107

[7] How to attack machine learning (evasion, poisoning, inference, trojans, backdoors). [Online]. Available: https://bit.ly/3iCBKmf

[8] A. Chan-Hon-Tong, "An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning," vol. 1, no. 1, pp. 192–204. [Online]. Available: http://www.mdpi.com/2504-4990/1/1/11

[9] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning." [Online]. Available: http://arxiv.org/abs/1712.05526

[10] A. K. Veldanda, K. Liu, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, B. Dolan-Gavitt, and S. Garg, "NNoculation: Catching BadNets in the wild," pp. 49–60. [Online]. Available: http://arxiv.org/abs/2002.08313

[11] Q. Xiao, K. Li, D. Zhang, and W. Xu, "Security risks in deep learning implementations," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, pp. 123–128. [Online]. Available: https://ieeexplore.ieee.org/document/8424643/

[12] Y. Ji, X. Zhang, S. Ji, X. Luo, and T. Wang, "Model-reuse attacks on deep learning systems." [Online]. Available: http://arxiv.org/abs/1812.00483

[13] R. Schuster, T. Schuster, Y. Meri, and V. Shmatikov, "Humpty dumpty: Controlling word meanings via corpus poisoning." [Online]. Available: http://arxiv.org/abs/2001.04935

[14] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," 2017.

[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning." [Online]. Available: http://arxiv.org/abs/1807.00459

[16] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction." [Online]. Available: http://arxiv.org/abs/1811.03604

[17] A. S. Rakin, Z. He, and D. Fan, "TBT: Targeted neural network attack with bit trojan." [Online]. Available: http://arxiv.org/abs/1909.05193

[18] S. Hong, P. Frigo, Y. Kaya, C. Giuffrida, and T. Dumitraș, "Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks." [Online]. Available: http://arxiv.org/abs/1906.01017

[19] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models." [Online]. Available: http://arxiv.org/abs/1610.05820

[20] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 739–753. [Online]. Available: https://ieeexplore.ieee.org/document/8835245/

[21] M. Ribeiro, K. Grolinger, and M. A. Capretz, "MLaaS: Machine learning as a service," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 896–902. [Online]. Available: http://ieeexplore.ieee.org/document/7424435/

[22] H. Yan, X. Li, H. Li, J. Li, W. Sun, and F. Li, "Monitoring-based differential privacy mechanism against query flooding-based model extraction attack," pp. 1–1. [Online]. Available: https://ieeexplore.ieee.org/document/9389670/

[23] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning." [Online]. Available: http://arxiv.org/abs/1802.05351