МЕХАНИЗМЫ ЗАЩИТЫ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ОТ СОСТЯЗАТЕЛЬНЫХ АТАК

DOI: 10.36724/2072-8735-2023-17-10-28-42

Manuscript received 12 August 2023;
Accepted 15 September 2023

Фомичева Светлана Григорьевна,
Санкт-Петербургский государственный университет аэрокосмического приборостроения, г. Санкт-Петербург, Россия, ***levikha@mail.ru***

Беззатеев Сергей Валентинович,
Санкт-Петербургский государственный университет аэрокосмического приборостроения, г. Санкт-Петербург, Россия, ***bsv@aanet.ru***

Ключевые слова: *модели машинного обучения, состязательная атака, механизмы защиты, оптимизация функции потерь, маскировка градиента, ансамбли моделей*

Проблемы защиты интеллектуальных информационных систем остро актуальны в силу их применения в субъектах критической информационной инфраструктуры. Наиболее трудными для выявления являются состязательные атаки на модели машинного обучения, которые осуществляются в ходе трансферного и федеративного обучения уже предобученных моделей. При этом в русскоязычном сегменте публикаций анатомия состязательных атак освещается редко, а механизмы защиты от них и механизмы оценки защиты от атак на модели машинного обучения практически отсутствуют, что актуализирует необходимость представленного в статье аналитического обзора. Целью исследования является проведение аналитического обзора и формализованное описание механизмов защиты моделей машинного обучения, которые являются целью состязательных атак. Результаты: опираясь на классификацию атак, направленных на модели машинного обучения, формализовано описаны современные принципы их механизмов защиты. В отличии от существующих публикаций в ходе нашего обзора выделены и обобщены не только виды атак на модели машинного обучения, но и механизмы их реализации. Проведена классификация существующих методов защиты от состязательных атак. Практическая значимость: в результате обобщения сформулированы необходимые требования к построению защищенных от состязательных атак моделей. Обсуждение: Основное внимание при построении механизмов защиты моделей машинного обучения и оценки их надежности должно быть сосредоточено на противодействие комплексным адаптивным атакам, которые явно выявляют и нацеливаются на самые слабые звенья защиты.

Информация об авторах:

Фомичева Светлана Григорьевна, *к.т.н., профессор, профессор Санкт-Петербургского государственного университета аэрокосмического приборостроения, г. Санкт-Петербург, Россия*

Беззатеев Сергей Валентинович, *д.т.н., доцент, заведующий кафедрой информационной безопасности Санкт-Петербургского государственного университета аэрокосмического приборостроения, г. Санкт-Петербург, Россия*

Для цитирования:
Фомичева С.Г., Беззатеев С.В. Механизмы зашиты моделей машинного обучения от состязательных атак // T-Comm: Телекоммуникации и транспорт. 2023. Том 17. №10. С. 28-42.

For citation:
Fomicheva S. G., Bezzateev S. V. Defenses for machine learning models from adversarial attacks. *High technologies in Earth space research. T-Comm*, vol. 17, no.10, pp. 28-42. (*in Russian*)

Введение

В настоящее время интеллектуальные информационные системы, решающие задачи классификации различного рода, получили повсеместное распространение. Однако при разработке мало задумывались об их потенциальной уязвимости, что на текущий момент стало глобальной проблемой. Новые технологии создали и целый спектр новых типов атак, нарушающих штатное функционирование корпоративных систем с интеллектуальными модулями принятия решений, в том числе и на объектах критической информационной инфраструктуры. Особенную озабоченность вызывают так называемые состязательные атаки на модели машинного обучения (ML-модели), которые осуществляются в ходе трансферного и/или федеративного обучения уже предобученных моделей. Это связывают с попытками автоматизировать разработку эксплойтов (специализированных программных решений, реализующих те или иные атаки на ML-модели) теми же средствами – методами машинного обучения. Следовательно, возникает проблема не только защиты систем искусственного интеллекта (AI), но и проблема защиты от искусственного интеллекта. При этом в русскоязычном сегменте публикаций анатомия атак на ML-модели освещается редко, а механизмы защиты от них и механизмы оценки защиты от атак на ML-модели практически отсутствуют.

Также следует отметить, что неконтролируемые попытки использования аугментации и/или генерация синтетических датасетов для увеличения обучающих выборок способны привести к неосознанному отравлению данных. Процесс модификации датасетов с целью атаки на ML-модель называют отравлением данных (data poisoning). Такие процессы требуется контролировать на этапах предобработки наборов данных, для чего нужны механизмы оценки защиты ML-модели.

В данной статье мы, опираясь на классификацию атак, направленных на ML-модели, формализовано анализируем механизмы защиты, которые являются целью состязательных атак. В отличии от существующих публикаций в ходе нашего обзора выделены и обобщены не только виды атак на ML-модели, но и механизмы их реализации, а также принципы, реализующие защиту ML-моделей.

1. Атаки на ML-модели

Не претендуя на полноту классификации существующих атак на ML-модели, выделим в данной работе те из них, которые относятся к состязательным или сопровождают состязательные атаки, условно разделив их на три основных вида в соответствии с этапами жизненного цикла ML-модели (рис. 1). Опираемся в данном случае на классификацию атак, предложенную в [1]:

1) *Атаки на процесс обучения*. Данный класс атак нацелен либо на обучающие и тестовые наборы данных (датасеты), либо на архитектуру самой ML-модели на этапе процесса обучения ML-модели.

Наличие отравленных обучающих экземпляров датасета (если они не маскируются с учетом механизмов защиты ML-модели) возможно обнаружить в ходе тестирования ML-модели. Как правило, метрики оценки ее качества (например, точность распознавания в задачах компьютерного зрения) существенно снижаются при наличии отравленных данных и имеют среднее или ниже среднего значения.

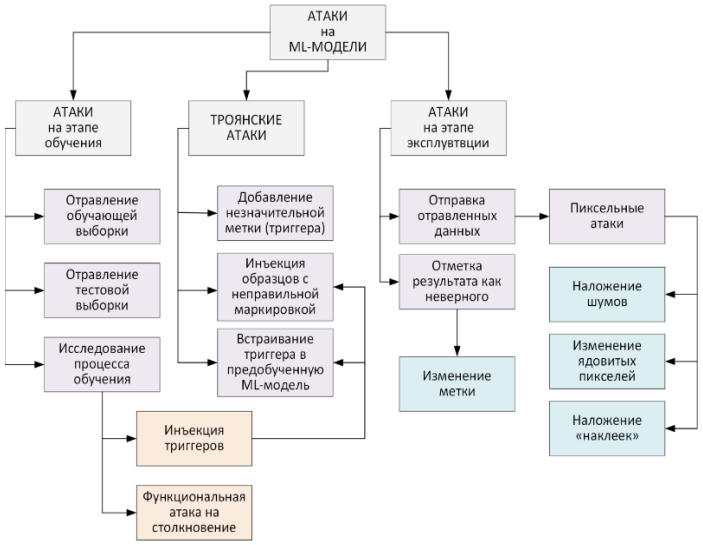

Рис. 1. Виды атак на ML-модели

2) *Троянские атаки***.** Данный класс атак основан на добавлении специальных триггеров («троянских оберток» и «троянских инъекций») в архитектуру ML-модели и метки-триггера, накладываемую на входной экземпляр данных модели. При этом ML-модель исправно функционирует до тех пор, пока не получит на вход экземпляр с меткой-триггером, в результате чего инициируется реакция, которая меняет поведение ML-модели в пользу атакующего,

Опасность такой атаки состоит в том, что на этапе тестирования не всегда можно обнаружить триггер. Такие атаки применяются при решении задачи по распознаванию и обнаружению объектов, при использовании генеративных моделей обучения, при обучении с подкреплением и для моделей с водяными знаками [2,3].

3) *Атаки при эксплуатации ML-модели*. Данному виду атак подвержены, как правило, модели при трансферном и федеративном обучении. Скомпрометированная ML-модель провоцируется на дообучение модели на измененных оптимизаторах и loss-функциях, что приводит к постепенному снижению качества модели вплоть до вывода ее из эксплуатации. Часто такие атаки называют адаптивными.

При реализации вышеперечисленных атак используется следующие возможности:

1) *Функции влияния* (Influence functions) [4] оценивают эффект бесконечно малого изменения тренировочных данных на параметры ML-модели в результате обучения, которые в дальнейшем используются при построении данных для обмана ML-модели. Атакующие используют такие функции влияния для создания более сильных отравляющих данных за счет улучшения двухуровневой оптимизации. Процесс использования подобных функций влияния называют маскировкой градиента.

2) *Переворачивание метки* (Label flipping) не изменяет и не создает данные датасета. При этом на выходе модели происходит подмена названия метки класса для входного экземпляра, что приводит к отравлению уже обученной модели. Данный способ атаки часто встречается при федеративном обучении [5, 6], то есть, когда данные (датасет) для обучения и тестирования изолированно распределены или присылаются извне.

3) *Использование "чистого" ответа* (Clean-label Attacks). Атакующий имеет доступ к процессу предварительной обработки экземпляров датасета, а именно при сопоставлении входного экземпляра (чистого или модифицированного) и ответа (метки класса) для обучающей выборки [7,8], в результате чего модель неправильно классифицирует целевой экземпляр после обучения на этих данных.

4) *Триггерная атака через «черный ход»* (blackdoor) – реализуется инъекция набора ложных данных, содержащими backdoor-триггер [7]. Атакующий вводит в тренировочную выборку для проведения атаки ложные образцы с неправильной маркировкой. Чтобы побудить модель полагаться на триггер, злоумышленник выбирает несколько естественных образов и неправильно маркирует их меткой целевого класса перед добавлением триггера [9]. Поэтому полученные входные экземпляры неправильно маркируются на основании их содержимого. Во время обучения модель в значительной степени полагается на простой в изучении триггер «черного хода», чтобы классифицировать эти изображения. В результате, когда триггер применяется к новому образу во время развертывания, модель присваивает ему целевую метку, как того хочет атакующий.

5) *Встраивание триггера в предварительно обученные модели* (Backdooring Pre-trained Models). Злоумышленник имеет доступ к уже обученной нейронной сети. Он может встроить триггер без полного переобучения и даже без доступа к обучающей выборке [10-12].

6) *Атаки при переобучении на новые задачи* (Attacks for Transfer Learning). Модель угроз такой атаки состоит из злоумышленника, который отправляет предварительно подготовленные данные для создания фонового экстрактора функций, но не имеет никакого контроля над процессом настройки обучения (переобучения) нейронной сети [13-14].

7) *Атаки при федеративном обучении* (Attacks for Federated Learning). Федеративное обучение страдает двумя большими проблемами, особенно трудно разрешимыми, когда на распределенных хостах имеется лишь маленькая выборка данных для обучения – низкая скорость и конфиденциальность обучения [15-17]. В частности, конфиденциальность может быть нарушена в ходе анализа исходных данных и выявления закономерностей работы классификатора.

8) *Функциональная атака на столкновение* (Feature collision) работает в настройках целевого отравления данных обучающей выборки, в которых целью является порча тренировочных данных так, чтобы происходила неправильная классификация целевого примера [18].

9) При реализации атак часто используют *двухуровневую оптимизацию* (Bilevel optimization), которая помогает портить данные обучающей выборки путем моделирования и последующей оптимизации прямого поиска подозрительных данных (состязательных примеров), которые приводят к повреждению моделей. Двухуровневая оптимизация дает оптимальный обучающий набор для атаки на простые модели [19-23].

10) *Вероятностное отравление* (p-Tampering) данных заключается в том, что с вероятностью p модифицируется бит изображения или целые образцы в обучающей выборке [24, 25]. Как правило, такой подход часто используется для создания отравляющего шума для входного изображения.

11) *Явление затухающего (исчезающего) градиента* (Vanishing gradients) обычно наблюдается в процессе обучения при использовании алгоритма обратного распространения ошибки. При расчете градиента ошибки loss-функции в ходе движения к начальным слоям нейросетевой модели фиксируются значения, стремящиеся к нулю, что сводит на нет эффективность процесса обучения [26].

12) *Явление взрывающегося градиента (Exploding gradients)* может также наблюдаться в случаях использования алгоритма обратного распространения ошибки, однако в отличие от явления затухающего градиента, наоборот, значения градиента стремительно растут, приводя к переполнению разрядности переменных, хранящих весовые коэффициенты нейронов [26].

2. Формальное представление процедур создания состязательных примеров

Перечисленные выше механизмы атак указывают, что подавляющая часть из них так или иначе использует отравление обучающего или тестового набора данных. Механизмы защиты против отравления наборов данных динамично прорабатываются [27 - 29]. Но их эффективность к сильным атакам, к числу которых относятся адаптивные атаки (т.е. атаки, которые были специально разработаны для нацеливания на данный (известный атакующему) механизм защиты) [30,31, попрежнему остро актуальны.

В [32] авторы отмечают, что технические инструменты для надлежащей оценки защиты ML-моделей уже существуют и более совершенную атаку можно построить, используя только те инструменты, которые хорошо апробированы. Из чего сделан вывод о необходимости разработки методологии использования существующих механизмов оценки защиты ML-моделей. При этом механизмы оценки зашиты должны быть сравнительно просты. Авторы отмечают, что атаки не обязательно должны быть сложными, даже при наличии защиты ML-модели. Они исходят из опыта проведенных экспериментов, показывающих, что в сложной защите часто имеется «слабое звено» из критически важных блоков ML-модели, нацеливание на которые позволяет нивелировать механизмы защиты достаточно простыми атаками. Чаще всего по своей сути простые атаки за счет возможности адаптации позволяют в автоматическом режиме проектировать такую loss-функцию, которую становится легче оптимизировать, чем нативные loss-функции ML-модели. И эта спроектированная loss-функция контролируется атакующим, позволяя проводить сильные атаки.

Отмечено, что, несмотря на возможность обобщения механизмов адаптации атак, ни одна стратегия атаки не является достаточной для всех механизмов защит. Следовательно, адаптивные атаки не могут быть в полной мере автоматизированы и всегда требуют соответствующей настройки для каждой системы защиты. Для проведения независимой переоценки предлагаемых средств защиты авторы разместили код с анатомией атак по адресу https://github.com/wielandbrendel/adaptive_attacks_paper.

Формализуем используемые механизмы защиты ML-моделей (рассматриваем классификационные ML-модели и их механизмы защиты на примерах состязательности).

Будем придерживаться обозначениям и терминам, предложенным для описания адаптивных атак в [32].

Для классификационной ML-модели f и доброкачественного входного сигнала x (например, полученного из тестового набора) с истинной меткой y примером состязательности [27, 32, 33] является возмущенный входной сигнал x', для которого при некоторой малой ε и некоторого выбранного класса $t \neq y$ выполняются соотношения (1):

$$\begin{cases} \|x - x'\|_p \le \varepsilon \\ f(x') \neq y, \text{ при нецелевой атаке} \\ f(x') = t, \text{ при целевой атаке} \end{cases} \quad (1)$$

Как было отмечено выше, в ходе состязательных атак одной из основных задач для атакующего является способность генерации состязательных сигналов. Для этого сначала требуется [32] построить альтернативную loss-функцию L такую, чтобы значение $L(x, y)$ было большим для $f(x) \neq y$ (обычно в качестве baseline loss-функции используется L_{CE} кросс-энтропия (https://digitrain.ru/articles/192991/). После этого атакующий старается максимизировать $L(x', y)$ при сохранении меры $\|x - x'\|$ малым значением.

Рассмотрим в качестве примера градиентный спуск по входному пространству [33], чтобы продемонстрировать формирование состязательных сигналов.

Пусть $x_0 = x$, а затем итеративно установим $x_{i+1} = \Pr oj\left(x_i + \alpha \cdot Normalize\left(\nabla_{x_i} L\left(x_i, y\right)\right)\right)$, там, где $Proj$ проецирует входные данные на меньшую область с незначительными искажениями, $Normalize$ нормализует шаг градиента в соответствии с рассматриваемой нормой, а α управляет размером шага. После N шагов (например, 1000) предполагается, что $x' = x$, y – результирующий пример состязательности.

В некоторых случаях полезно ML-модель, являющуюся классификатором, описать в виде $f(x) = arg\ max\ z(x)$, где $z(x)$ – это вектор оценок класса, который в [32] называют логитом.

Пусть все анализируемые средства защиты являются «белым ящиком», то есть атакующий обладает знаниями об архитектуре модели, весах и любой необходимой дополнительной информации об архитектуре ML-модели. Приведем описание общих стратегий атаки, используемых при оценке защиты ML-моделей.

PGD. Projected Gradient Descent - *Прогнозируемый градиентный спуск* [33] использует ранее введенную в рассмотрение норму $\|x - x'\| \le \varepsilon$, чтобы реализовать поиск состязательного сигнала x'.

Пусть l_p означает лебегово пространство измеримых функций. Будем рассматривать гиперокрестность D в l_p-пространстве с заданным радиусом $r \le \varepsilon$, центром которой является x. Выбирая случайную начальную точку $x_0 \in D$, атакующий итеративно выполняет действия по максимизации функции потерь при заданном ограничении на возмущение сигнала в соответствии с выражением (2):

$$\begin{cases} x_{i+1} = Projection_D\left(x_i + step \cdot g\right) \\ for\ g = \underset{\|D\|_p \le 1}{\arg\max}\left(D^{\mathrm{T}} \nabla_{x_i} L\left(x_i, y\right)\right) \end{cases} \quad (2)$$

где $L(x, y)$ – подходящая функция потерь (например, кросс-энтропия), $step$ - размер шага итерации, $Projection_D$ – оператор (например, оператор отсечения), который проецирует входные данные на D, а g – направление наивысшего подъема для данной l_p-нормы.

C&W. (Carlini & Wagner) В работе [31] авторы предлагают в ходе состязательной атаки вместо максимизации функции потерь $L(x,y)$ (альтернативно стратегии PGD), производить поиск наименьшего успешного противоборствующего возмущения $\|x - x'\|_p$ в соответствии с выражением (3):

$$\left(L\left(x', y\right) - \lambda \cdot \|x' - x\|_p\right) \xrightarrow{x'} \max. \quad (3)$$

Бинарного поиск решения (3) по параметру $\lambda > 0$ позволяет найти баланс, где максимизируется $L(x', y)$ и минимизируется $\|x - x'\|$, что в результате приводит к успешной атаке.

BPDA (Backward Pass Differentiable Approximation) [30] – Стратегия атак, называемая *дифференцируемой аппроксимацией с обратным проходом,* является разновидностью градиентной аппроксимации, но, в отличие от PGD и C&W, направлена на средства защиты с одним или несколькими не дифференцируемыми компонентами.

Если представить ML-модель $f(x)$ как структуру из n-взаимосвязанных компонент $f(x) = f_n \circ f_{n-1} \circ \cdots \circ f_i \circ \cdots \circ f_1(x)$, где компонент f_i является не дифференцируемым, то идея BPDA-стратегии заключается в поиске апроксимизирующей *дифференцируемой* функции $g(x)$ такой, что $g(x) \approx f_i(x)$. Часто просто выбирают $g(x) = x$. При этом при прямом проходе алгоритма обратного распространения ошибки реализуют вычисление, по-прежнему, $f(x) = f_n \circ f_{n-1} \circ \cdots \circ f_i \circ \cdots \circ f_1(x)$, а при обратном проходе реализуют $f(x) = f_n \circ f_{n-1} \circ \cdots \circ g_i \circ \cdots \circ f_1(x)$. Поскольку атакующий знает $g(x)$, то он реализует стандартную целевую атаку именно на этот компонент.

EOT (Expectation over Transformation). *Ожидание над преобразованием* [34] – метод вычисления градиентов моделей с рандомизированными компонентами. Идея EOT предполагает получение градиентов математического ожидания любой случайной функции. Учитывая рандомизированный классификатор $f_r(x)$ (где r обозначает внутреннюю случайность классификатора), можно вычислить:

$$\nabla_x \mathrm{E}_r\left[f_r(x)\right] = \mathrm{E}_r\left[\nabla_x f_r(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_x f_{r_i}(x), \quad (4)$$

где r_i – независимые значения случайной функции. Как и в случае с BPDA, аппроксимированные градиенты затем могут быть подключены к любой стандартной атаке.

Кроме того, в [32] сформулировано семь основных постулатов построения эффективных адаптивных атак:

I. *Стремление к простоте.* Атака должна быть максимально приближена к прямому градиентному спуску с соответствующей функцией потерь. Дополнительные усложнения следует вводить только тогда, когда более простые атаки терпят неудачу. Это обусловлено тем, что легче диагностировать сбои в более простых атаках, чтобы после их нивелирования перейти к более сильной атаке.

II. *Атака (функция, близкая к) полной защиты.* Если механизм защиты поддается сквозной дифференциации, то атаке должен подвергаться весь механизм защиты целиком. Следует избегать дополнительных условий в функциях по-

терь, если в них нет необходимости; и наоборот, по возможности следует включать любые компоненты, особенно функции предварительной обработки.

III. *Определение важных компонент защиты и нацеливание на них*. Некоторые средства защиты сложным образом сочетают в себе множество подкомпонентов. Часто проверка этих компонентов показывает, что для обхода защиты достаточно только одного или двух. Нацеливание только на эти компоненты может привести к более простым, действенным и легко оптимизируемым атакам.

IV. *Упрощение атаки через адаптацию цели*. Способность генерации состязательных сигналов проще реализуется при наличии у атакующего контролируемой им loss-функции. Хотя процесс формирования таких loss-функций не тривиален, но его успешная реализация значительно повысит вероятность атаки.

V. *Проверка постоянности функции потерь*. Наилучший оптимизатор, способный гарантированно находить оптимальное значение loss-функции, не является гарантией успешной атаки,

VI. *Оптимизация функции потерь с помощью различных методов*. Следует выбирать функцию потерь простую, насколько это возможно, а также следует выбрать правильные гиперпараметры (например, достаточное количество итераций или повторений) [36, 37].

VII. *Использование сильной адаптивной атаки для обучения в состязательном режиме*. Многие комбинированные средства защиты приводят к меньшей надежности, чем исключительно состязательное обучение. Это показывает еще один способ сбоя слабых адаптивных атак: если атака, используемая для обучения, не позволяет надежно найти примеры соперничества, модель не будет противостоять более сильным атакам.

3. Механизмы защиты ML-моделей от состязательных атак

Проанализировав механизмы состязательных атак, их стратегии реализации и постулаты, а также механизмы противодействия им, нами были выявлены следующие подходы к построению систем защиты ML-моделей:

1) Статистические тесты
2) Методы восстановления входного образа
3) Маскировка градиента
4) Модификация исходной функции потерь
5) Формирование защитных ансамблей ML-моделей
6) Тренинг на состязательных примерах
7) Стохастическая интерполяция

В таблице 1 представлены проанализированные механизмы защиты ML-моделей.

Приведем в формализованном виде формулировку этих подходов с указанием достоинств и потенциальных уязвимостей.

3.1. Статистические тесты

3.1.1. The Odds are Odd

В статье [38] предлагается выявлять состязательные атаки с помощью статистических тестов, основанных на анализе распределения значений логитов классификатора. Тест исходит из предположения, что состязательные образцы менее устойчивы к шуму, чем доброкачественные. А именно, механизм защиты формирует для заданного входного сигнала x логит-вектор $z(x)$ сначала без его предварительного искажения, а затем накладывает шум на x из некоторого фиксированного распределения N (например, гауссовского), для которого также находят вектор логитов $z(x + \delta)$, где $\delta \sim N$. Предполагается, что для доброкачественных примеров $z(x) \approx z(x + \delta)$, в то время как для состязательных примеров два логит-вектора будут существенно отличаться.

Однако, по утверждению [32] гипотеза, что особенности состязательных примеров, созданных стандартными атаками, могут быть использованы для обнаружения всех состязательных примеров, является заблуждением. Например, для модели логистической регрессии устойчивость к шуму и достоверность модели напрямую связаны: - достоверность модели зависит от расстояния входных данных до границы принятия решения, что, в свою очередь, определяет устойчивость входных данных к (гауссовскому) шуму.

3.1.2. Превращение слабости в силу

Механизм защиты «Weakness into a Strength» [39] является примером обнаружения враждебности путем комбинирования функции потерь и статистического модуля. В то время как идея защиты концептуально проста, исходная методология оценки защиты сложна, и построение единой унифицированной функции потерь, которая кодирует цель атакующего, является сложной задачей.

Обозначим через $f(x)_i$ вероятность того, что классификатор отнесет входной сигнал x к классу i. Чтобы определить, является ли данный входной сигнал x состязательным, механизм защиты устанавливает пороговые значения r_1 и r_2 и проверяет, верно ли любое из следующих значений

$$\begin{cases} \sum_i \left\| f(x+\delta)_i - f(x)_i \right\|_1 \geq r_1 \ \text{ for } \delta \sim \mathrm{N}\left(0, I\sigma^2\right), \text{ или} \\ \#Steps(f, x) > r_2 \end{cases} \quad (5)$$

где #*Steps*- число шагов, потребовавшихся для генерации состязательного примера x', такого, что $f(x) \neq y$ или $f(x)=t \neq y$.

Верхний член в (5) означает, что предсказания моделей не должны существенно изменяться, когда входные данные слегка возмущены гауссовским шумом. Нижний член в (6) является статистическим и более сложен в понимании: учитывая входные данные x, вычисляется количество необходимых шагов для превращения $x \rightarrow x'$. То есть если количество шагов слишком велико от границы принятия решения, то этот тест завершится неудачей.

Построение адаптивной атаки против механизма защиты (5) проводится путем разработки новой функции потерь $L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \lambda_4 L_4$, которая определяется как комбинация четырех независимых членов loss-функции, где λ_i –весовые коэффициенты. Первый член L_1 - стандартная кросс-энтропия, задача которой «обмануть» классификатор, $L_1 = L_{CE}(f(x); y_t)$. Второй член L_2 побуждает классификатор быть уверенным в своем прогнозе путем прямой оптимизации первого члена обнаружения: $L_2 = \mathrm{E}_{\delta \sim \mathrm{N}(0, I\sigma^2)} \left[\left\| f(x) - f(x'+\delta) \right\|_1 \right]$, чтобы обойти первый детектор. Третий член L_3 поощряет классификатор к высоким потерям после выполнения одного шага градиентного спуска к любому неправильному классу:

$$L_3 = -\mathrm{E}_{y' \neq y_t} L_{CE}\left(f\left(x - \alpha \nabla_x L_{CE}\left(f(x), y'\right)\right), y'\right).$$

Таблица 1

Подходы и механизмы защиты ML-моделей от состязательных атак

Механизм защиты	Гипотеза	Авторы защиты, год	Эффективность к стратегии атак					
			PGD	C&W	BPDA	EOT	PGD+ EOT	Adaptive attack
Статистические тесты								
The Odds are Odd	Состязательные примеры менее устойчивы к шуму, чем доброкачественные примеры	Roth et al., 2019	+	+	-	-	-	-
Weakness into a Strength	Комбинирование функции потерь и статистического модуля повысит надежность защиты	Hu et al., 2019	+	+	-	+	+	-
Маскировка градиента								
(kWTA	Квантование функций активации классифицирующего слоя повысит надежность защиты	Xiao et al., 2020	+	+	+	-	-	-
Тренинг на состязательных примерах								
Generative Classifier	Превентивное обучение на состязательных примерах повысит надежность защиты	Li et al., 2019	+	+	-	+	+	-
Asymmetrical Adversarial Training	Модели, обученные состязательности, лучше выявляют примеры состязательности	Yin et al., 2020	+	+	+	+	+	+
Методы восстановления входного образа								
Robust Sparse Fourier Transform	Спектральное преобразование и последующее восстановление изображения повысит точность классификации	Bafna et al., 2018	+	-	+	+	+	-
ME-Net	Стохастическое прореживание пикселей и последующее восстановление изображения повысит точность классификации	Yang et al., 2019a	+	+	+	+	+	-
Формирование защитных ансамблей ML-моделей								
Error Correcting Codes	Корректирующий код тренирует классификатор решать дополнительную задачу, а дополнительное разнообразие затруднит атакам поиск состязательных примеров	Verma & Swami, 2019	+	+	+	+	+	+
Ensemble Diversity	По сравнению с одиночными моделями или менее разнообразными ансамблями дополнительное разнообразие затруднит атакам поиск состязательных примеров	Pang et al., 2019	+	+	+	+	+	-
EMPIR	Построения ансамбля моделей со смешанной точностью весов и активаций затруднит атакам поиск состязательных примеров	Sen et al., 2020	+	+	+	+	+	-
Модификация исходной функции потерь								
Rethinking Softmax Cross Entropy	Замена исходной функции потерь затруднит формирование состязательных примеров	Peng et al., 2020a	+	+	+	-	-	-
Стохастическая интерполяция								
Mixup Inference	Конкурирующее возмущение имеет тенденцию оказывать влияние на модель только вблизи исходного образа и что большие сдвиги от него ослабляют влияние на модель	Peng et al., 2020b	+	+	+	+	+	-

Четвертый член L_4 поощряет то же самое, но для целевого класса y_t:

$$L_4 = -L_{CE}\left(f\left(x+\alpha\nabla_x L_{CE}\left(f(x), y_t\right)\right), y_t\right). \quad (6)$$

Функции потерь L_3 и L_4 не дифференцируемы, и поэтому в статье применяется BPDA-стратегия [34]. Комбинированная функция потерь (5) исключительно сложна, и маловероятно, что ее оптимизация позволит свести к минимуму все четыре члена одновременно.

Основное предположение, на которое опирается данный механизм защиты, заключается в том, что не существует входных данных, которые (а) имеют высокую степень достоверности в отношении случайного шума, но (б) все еще близки к границе принятия решения при проведении состязательной атаки.

В результате анализа механизмов защиты на базе статистических тестов, можно сделать ряд промежуточных выводов:

1) Всегда существует адаптивная атака, способная генерировать состязательные образцы, которые так же близки к границам принятия решений и обладают такой же достоверностью, как и доброкачественные примеры.

2) Объединение нескольких термов (членов) loss-функции приводит к трудно оптимизируемым функциям потерь, которые могут вести себя не так, как хотелось бы атакующему.

3) Стратегию BPDA не следует рассматривать как метод общего назначения для минимизации с помощью произвольных

недифференцируемых функций. Скорее всего, функции потерь должны быть специализированно разработаны для работы с BPDA и уже должны быть практически дифференцируемыми.

3.2. Маскирование градиента

3.2.1. k-победителей забирают все (k-Winners Take All)

Механизм защиты k-Winners-Take-All (kWTA) преднамеренно затрудняет спуск по градиенту. В статье [40] с данной целью предлагается функция активации, которая намеренно предназначена для маскировки обратного распространения градиентов для защиты от атак. Предложенный механизм защиты заменяет стандартную функцию активации ReLU в ML-модели (CNN) дискретной функцией k-WTA (7), которая модифицирует выходные данные слоя классификации:

$$\varphi_k(\boldsymbol{y})_j = \begin{cases} y_i, y_i \in \{k - \text{наибольших элементов } \boldsymbol{y}\} \\ 0, \text{ в остальных случаях} \end{cases}, \quad (7)$$

Число «победителей» в (7) для каждого из *j*-слоев определяется своим (различным) параметром *k*, который получают помощью процедуры прореживания выходных данных слоя классификации [40]. В результате, если атакующий в ходе поиска состязательного сигнала пытается даже незначительно изменить входные данные, механизм защиты кардинально меняет выходы функций активации и, следовательно, приводит к большим скачкам в прогнозах и уничтожает всю полезную информацию о градиенте.

К применению в механизмах защиты стратегии k-WTA следует подходить с осторожностью. В частности, если в структуру любой незащищенной ML-модели добавить компонент с простой не дифференцируемой функцией активации, которая чрезвычайно точно квантует логиты, то наивные атаки, основанные на градиенте, станут безуспешными в силу неопределенности градиентов. Однако оценка градиента по конечным разностям все равно будет работать (например, атака BPDA [34] может рассматриваться как пример такого подхода, или использование обратного нейро-нечеткого вывода [40-43] на базе алгоритмов мягкого квантования знаний позволят восстановить градиент со сколь угодно малой точностью). В качестве сценария успешной атаки на механизм защиты k-WTA можно использовать «усреднение случайности», то есть оценивать средний локальный градиент δ в точке *x* с достаточно большим числом случайных нормально распределенных возмущений (например, *M* = [15000, 20000], со стандартным отклонением ε = 8/255), т.е.,

$$g(x) = \frac{2}{\sigma M}\sum_{j=0}^{M/2}\left[\nabla_x L_{CE}\left(f(x+\delta), y\right) + \nabla_x L_{CE}\left(f(x-\delta), y\right)\right], \quad (8)$$

где $\delta \sim N(\mu = 0, \sigma^2)$, а ∇L_{CE} - потеря кросс-энтропии. Авторы эксплойта данной атаки [32] за 100 итераций успешно сгенерировали состязательные примеры для обученных состязаниям моделям ResNet-18, которые достигают точности 0,16%.

Следовательно, k-WTA не только не является эффективной защитой, но и ухудшает подготовку к состязаниям. Кроме того, в [40] теоретически обосновывается, а в [42, 44] приводятся реализации прямого и обратного преобразования нейро-нечетких структур (принятии решений) при квантовании знаний (в данном контексте под знанием следует понимать конкретную закономерность, задаваемую функций потерь). Что говорит о том, что атаки, основанные на принятии решений (деревья решений), могут работать так же хорошо или даже лучше, чем атаки, основанные на градиентах, в частности, когда градиенты (намеренно) маскируются.

3.3. Тренинг на состязательных примерах

Основная идея состязательного обучения с точки зрения задач защиты ML-моделей состоит в том, чтобы предварительно создать пул состязательных примеров, после чего включить эти состязательные примеры в процесс обучения ML-модели с соответствующей разметкой, позволяя тем самым модели различать состязательные и доброкачественные примеры при обучении.

Чтобы сформировать состязательный пример, ML-модель вынуждают вместо минимизации функции потерь максимизировать потери *L*, то есть решить задачу оптимизации:

$$L\left(h_\theta(x'), y\right) \xrightarrow{x'} \max, \quad (9)$$

где $h_\theta(\cdot)$ – гипотетическая функция ML-модели, *x'* – состязательный пример, *y* – метка истинного класса *x*.

Обычно для реализации (9), оптимизируют возмущение δ для *x:*

$$L\left(h_\theta(x+\delta), y\right) \xrightarrow{\delta \in \Delta} \max, \quad (10)$$

где Δ – допустимый набор возмущений: $\Delta = \left\{\delta : \|\delta\|_\infty \le \varepsilon\right\}$, а $\|\delta\|_\infty = \max_i |\delta_i|$.

Выражения (9) и (10), по сути, соответствуют нецелевой атаке. А целевая атака должна не только максимизировать выражение (10), но и минимизировать loss-функцию целевого класса:

$$L\left(h_\theta(x+\delta), y\right) - L\left(h_\theta(x+\delta), y_{ttargett}\right) \xrightarrow{\delta \in \Delta} \max. \quad (11)$$

Тогда в общем виде задача обучения классификатора, устойчивого к враждебным атакам, сводится задаче минимизации эмпирического враждебный риска R_{adv} на обучающем наборе данных D_{train}, задаваемого выражением (12):

$$\min_\theta R_{adv}(h_\theta, D_{train}) \equiv \min_\theta \frac{1}{|D_{train}|}\sum_{(x,y)\in D_{train}} \max_{\delta\in\Delta}\left(L\left(h_\theta(x+\delta), y\right)\right). \quad (12)$$

3.3.1. Механизмы защиты на базе генеративных классификаторах

Механизм защиты на базе генеративных классификаторах (Generative Classifier) [45] агрегирует многие характеристики состязательного обучения, которые затрудняют оценку защиты. В нем используются несколько ML-моделей, несколько типов loss-функций, стохастичность и дополнительный этап обнаружения. Структурно данный механизм защиты в качестве baseline-модели использует вариационный автоэнкодер. В частности в [45], авторы предполагают, что экземпляры набора данных *(x, y)* генерируются неким неизвестным процессом со скрытой переменной η. В ходе обучения автоэнкодера они восстанавливают различные совместные распределения $p(x, y, \eta) = p(\eta)\, p(y|\eta)\, p(x|\eta)$.

Классификатор (Алгоритм 1) объединяет три ML-модели (CNN): кодер *enc* и два декодера *dec1*, *dec2*. Кодер для каждого класса генерирует параметры для выборки случайных

скрытых векторов η. По этим скрытым векторам декодер *dec1* восстанавливает входные данные, а декодер *dec2* - метку класса. Механизм защиты запускает эти три ML-модели для N случайных скрытых векторов (по умолчанию $N = 10$) для каждого класса, и вычисляет четыре оценки, которые в совокупности формируют логит $z(x)=[l_1 \ \dots \ l_K]$. Механизм защиты опирается на генеративный подход, который часто постулируется как более надежный, чем чисто дискриминационные модели защиты.

Алгоритм 1. **Генеративный классификатор** (Generative Classifier, [45])

Input: Data point x
Output: Logits $z(x)$
For $k \in [1,K]$ do
μ, σ = **enc**(x,k)
For $i \in [1,N]$ **do**
$\eta \leftarrow$ Ń$(\mu, \sigma \cdot \boldsymbol{I})$
x^*=**dec**$_1(\eta)$
y^*=**dec**$_2(\eta)$
$L_{recons}=||x-x^*||^2_2$
L_{CE}=cross-entropy (y^*,k)
P_{prior}=log Ń$(\eta;0, \boldsymbol{I})$
$P_{posrerior}$=log Ń$(\eta; \mu, \sigma \cdot \boldsymbol{I})$
$score_i=- L_{recons} - L_{CE} + P_{prior} - P_{posrerior}$
end
$l_k=\log((1/N)\Sigma^N_{i=1} \exp(score_i)$
end
return $z(x)=[l_1 \ \dots \ l_K]$

Сложность этого механизма защиты иллюстрирует преимущество сосредоточения на простых атаках, которые нацелены на наиболее важные элементы защиты. Механизм защиты представляет собой комплекс, который включает в себя три сети, четыре оценочных условия, множество комбинаций экспонент и логарифмов, интенсивную рандомизацию и этап обнаружения. Таким образом, существующие атаки (например, PGD [34] или C&W [31]) оказываются неэффективными: эффективное сочетание нескольких условий функции потерь никогда не бывает легким для атакующего. Однако в [32] выявлено, что все четыре составляющих комбинированной функции потерь, из которых складывается результирующая оценка защиты, не всегда имеют одинаковую важность. Если атакующий будет достаточно сконцентрирован на одной из них, то ему становится проще оптимизировать альтернативную функцию потерь. В частности, показано, что защита уязвима для "тривиальной" атаки, которая непрерывно вводит одни и те же входные данные до тех пор, пока модель случайно не классифицирует их неправильно. Кроме того, было выявлено, что L_{CE} – это единственный важный член комбинированной функции потерь. Остальные члены мало различаются в зависимости от оценок в классе.

3.3.2. Асимметричное состязательное обучение

В статье [46] рассматривается задача K-классовой классификации, где в качестве механизма защиты используется K моделей детекторов $h_1; \dots ; h_K$. Каждый детектор для заданного входного x формирует логит-оценку $h_i(x) \in R$ для соответствующего класса i. Детекторы обучаются по принципу состязательности в соответствии с (12), решая минимаксную задачу так, что для каждого обучающей пары (x,y) максимизируется $(h_y(x))$ и минимизируются составляющая $\max_{\|\delta\| \le \varepsilon} \sigma\left(h_i\left(x+\delta\right)\right)$ для всех $i \neq y$, где $\sigma(\cdot)$ обозначает сигмовидную функцию активации. Максимизируемый член (также называемый базовым детектором) гарантирует, что детектор правильного класса y распознает x как доброкачественный. Минимизируемые составляющие побуждают детекторы остальных классов отклонять возмущенные версии x.

По сравнению со многими другими средствами защиты, асимметричное состязательное обучение в виде набора детекторов, обученных противодействию, структурно довольно просто и не маскирует градиенты. Хотя в целом, при правильном выборе функций потерь для детекторов данный механизм защиты вполне надежен, вариант предложенных в [46] функций потерь имеет уязвимости. Так в [32, 47] выявлена проблема неоптимального выбора функции потерь при рассмотрении адаптивной атаки. Конкретизируем выявленную уязвимость.

Обозначим через $z(x)_i$ логит i-ого класса базового классификатора. Для каждой входной пары (x,y), атакующий при воздействии на интегрированный классификатор пытается найти состязательный сигнал x', который неправильно классифицирован базовым классификатором, в соответствии с выражением (12):

$$L\left(x',y\right)=\begin{cases}\max\limits_{i\neq y} z\left(x'\right)_i - z\left(x'\right)_y, & \text{если } f\left(x'\right)=y \\ \max\limits_{i\neq y} h_i\left(x'\right), & \text{иначе}\end{cases} \quad (13)$$

После того, как базовый классификатор обманут, атакующий «страхует» себя от отклонения найденного состязательного сигнала x', дополнительно максимизируя оценку всех детекторов, отличных от истинного класса y.

Проблема с этой функцией потерь заключается в том, что второй член, $\max_{i\neq y} h_i(x)$, не отражает истинную цель атаки. Действительно, чтобы обойти защиту, нужно всего лишь обмануть детектор данного класса прогнозировать базовым классификатором, т.е. $y' = f(x')$. Таким образом, максимизация количества всех детекторов приводит к расточительству ограниченного бюджета возмущений при атаке. Как показано в [32], данная стратегия атаки (13) снизила точность ML-модели, но не привела к полному снижению точности ни одной из моделей до 0%. Устойчивость механизма защиты обоснована следующим:

- Простотой механизма защиты, которая не предполагает каких-либо проблем с маскировкой градиента.
- Высокой интерпретируемостью механизма защиты.
- Устойчивостью средств защиты при стандартных проверках [48]: увеличение числа повторения PGD (до 200) и случайных перезапусков (до 10) снижают точность защиты только на дополнительный 1%.

Для механизмов защиты с асимметричным состязательным обучением характерно:

1) Хорошая функция потерь с точки зрения атакующего обладает тем свойством, что увеличение потерь всегда увеличивает вероятность успеха атаки. Функции потерь без этого свойства могут "растрачивать" бюджет возмущений впустую.

2) Иногда легче записать хорошую целевую функцию потерь, чем нецелевую. "Многоцелевая" атака, которая по очереди нацелена на каждый класс, в таком случае будет более эффективной, чем нецелевая атака.

В результате, при формировании механизмов защиты с помощью тренинга на состязательных примерах требуется:

1) Для комплексной защиты разложить вклад отдельных этапов классификации, чтобы определить, на каких из них следует сосредоточить внимание.

2) Выполнить оценку защиты от случайного шума.

3) Учесть возможность влияния альтернативных функций потерь (функций-противников). Функции-противники полезны для совместного обхода классификатора и детектора, поскольку они гарантируют, что любая статистика, вычисленная поверх функций-противников, соответствует статистике из чистых примеров.

3.3. Методы восстановления входного образа

3.3.1. Надежное разреженное преобразование Фурье

Существует ряд механизмов защиты ML-моделей от состязательных атак, в основе которых используются методы снижения размерности с последующим восстановлением поврежденных входных данных. Так в статье [49] представлен механизм защиты от l_0 –враждебного шума с помощью “надежного разреженного преобразования Фурье” (RSFT - Robust Sparse Fourier Transform). Предлагается “сжимать” каждое входное изображение путем проецирования на верхние *k* коэффициентов дискретного косинусного преобразования. Затем, сжатый образ восстанавливают и повторно классифицируют восстановленное изображение.

Процесс обучения ML-модели проводят на сжатых изображениях ло тех пор, пока не достигнута требуемая точность классификации как до, так и после преобразования. В [49] используется при сжатии входных данных метод итеративного жесткого порогового значения (IHT) с помощью преобразования Фурье и последующем восстановлении инвертированных входных данных. Хотя состязательные примеры в базовом классификаторе, предварительно обработанные подобным методом защиты, впоследствии больше не классифицируются базовым классификатором как состязательные, в [48] утверждается, что этого недостаточно для демонстрации надежности механизма защиты.

3.3.2. ME-Net

В статье [50] предлагается метод защиты, использующий матричную оценку (ME). Первоначально выполняется этап предварительной обработки входного изображения, в ходе которого случайным образом отбрасывается достаточно большая часть пикселей в изображении, а затем. используется ME-метод для восстановления изображения. Механизм защиты обучает ML-модель на таких предварительно обработанных входных данных с целью изучения представлений, которые менее чувствительны к небольшим вариациям входных данных. При этом изображение *x* представляется в виде матрицы *M*, а механизм защиты сначала отбрасывает каждую запись в *M* независимо с некоторой вероятностью *p*, чтобы получить зашумленную матрицу *N*. Затем он восстанавливает матрицу $\hat{M}$ из *N*, которая ожидаемо должна быть близка к *M*. Авторы исходят из предположения, что данный процесс разрушает структуру враждебного шума, одновременно усиливая глобальную структуру исходного изображения.

Для обучения ML-модели генерируются *n* матриц со случайным шумом $\hat{M}^{(1)},,,\hat{M}^{(n)}$ для каждого входного *x*. Эти матричные представления добавляются в исходный датасет и проводится дообучение стандартного классификатора.

Комбинируя ME-Net с механизмами защиты состязательного обучения существенно повышается устойчивость к состязательным атакам. Следует отметить, что предварительная обработка в ME-Net защите не поддается дифференцированию, что делает данный механизм зашиты устойчивым к атаке "белого ящика".

В целом при разработке механизмов защиты на основе восстановления входных данных следует:

1) Комбинировать функцию потерь и не дифференцируемую функцию предварительной обработки изображения (Например, некоторое разреженное спектральное преобразование и/или маскировка фрагмента изображения).

2) Помнить, что атакующий учитывает случайность на уровне всего механизма защиты.

3) Обучение базовых классификаторов дополнительным компонентам, исследующих удаление/восстановление фрагментов изображения позволяет выявить, источник повышения надежности.

3.4. Модификация исходной функции потерь

Большинство ML-моделей на базе глубоких нейронных сетей (DNN) структурно состоят из последовательности сверточных слоев, осуществляющих нелинейное преобразование входных данных в некоторое представление их признаков. Конечным полносвязным слоем, как правило, является линейный классификатор Softmax Regression (SR) при многоклассовой классификации и Logistic Regression (LR) при бинарной классификации. Повысить устойчивость к враждебным состязательным атакам могут позволить альтернативные правильно спроектированные классификаторы заключительного слоя.

В статье [51] предлагается на этапе классификации вместо SR или LG использовать процедуру линейного дискриминантного анализа Макса-Махаланобиса (MM-LDA), чтобы решить задачу адаптации доменов. В данном случае адаптация доменов необходима, чтобы привести распределение входных данных к совокупности гауссовских распределений (обратное направление процедуры также верно). В частности, в своем механизме защиты авторы меняют стандартный слой softmax и кросс-энтропию на loss-функцию в виде метрики Max-Mahalanobis Classifie (MMC), определяемую следующим выражением,

$$L_{MMC}\left(f^*(x),y\right)=\frac{1}{2}\left\|f^*(x)-\mu_y\right\|_2, \qquad (14)$$

где $f^*(x)$ – функция-экстрактор, $\mu_y \in R^N$ – центроиды распределения Max-Mahalanobis (MMD.

ML-модель классифицирует входной сигнал *x* в один из *K*-классов, учитывая обученную (адаптивную) функцию извлечения признаков f^* по правилу:

$$f(x)=\arg\min_{1\le i\le K}\left\|f^*(x)-\mu_i\right\|_2. \qquad (15)$$

Основная эвристика предложенной выше процедуры связана с тем, что SR- сети явно не моделируют распределение логитов, в то время как MM-LDA моделирует его как хорошо структурированный MMD. В результате при обучении ML-модели процедура MM-LDA влияет на сеть нелинейного преобразования посредством обратного распространения

ошибки, увеличивая минимальное расстояние между центроидами логитов, что в свою очередь, снижает риск враждебной состязательной атаки.

Существует ряд средств защиты, которые направлены на замену стандартной кросс-энтропии в слое softmax альтернативными функциями. Всякий раз, когда механизм защиты изменяет функцию потерь (например, как в (14)), используемую для обучения модели (15), исходят из предположения, что новая модель имеет поверхность функции потерь, которая хуже подходит для атак, нежели поверхность со стандартной кросс-энтропией.

Данный подход продолжают исследовать. В [32] рекомендовано каждый раз, когда в целях защиты изменяется функция потерь, используемая во время обучения, всегда следует тщательно изучать эту функцию потерь, с точки зрения ее атаки.

3.5. Формирование защитных ансамблей ML-моделей

3.5.1. Коды исправления ошибок при ансамблировании ML-моделей

Ансамбли ML-моделей, использующие в своей основе коды исправления ошибок (ECOC), изначально использовались для решения проблем масштабирования алгоритмов бинарной классификации в алгоритмы многоклассовой классификации. Основная идея заключалась в представлении многоклассовой задачи классификации в структуру с фиксированным числом задач бинарной классификации, при котором коды с исправлением ошибок позволяют кодировать каждый класс как произвольное количество задач бинарной классификации. То есть проблема классификации с несколькими классами переформатируется в проблему с множественной бинарной классификацией. В рассматриваемом случае дополнительные модели ансамбля работают как прогнозы «исправления ошибок», что приводит не только к повышению эффективности классификации, как таковой, но и делают ансамбль устойчивым к состязательным атакам.

В данных моделях каждому классу приписывается кодовое слово длины, превышающей количество бит, необходимое для уникального представления метки каждого класса, намеренно вводя избыточность.

Во время обучения каждую модель ансамбля при получении полного входного шаблона тренируют предсказывать только одну позицию в выходной строке. При эксплуатации ансамбля для новых входных данных используют каждую модель, чтобы сделать прогноз для создания двоичной выходной строки, а затем сравнивают двоичную строку с известным кодовым словом каждого класса [52]. В качестве выходных данных выбирается класс, который имеет наименьшее расстояние до предсказанного. Минимальное кодовое расстояние корректирующего кода является ключевым гиперпараметром ML-модели, определяющим ее устойчивость к враждебным атакам,

Очевидно, что, если классификаторов недостаточно, то нельзя однозначно восстановить результат, учитывая выходные данные моделей (в частности, нужно, по крайней мере, для K классов иметь значение $M \geq \log(K)$). В частности, каждый классификатор обучается как функция $z_i : X \rightarrow R$, а затем $f_i(x) = \text{sigmoid}\ (z_i(x))$. Чтобы сгенерировать окончательные прогнозы, определяется вектор $Z(x) = [z_1(x), z_2(x), ..., z_M(x)]$, а затем возвращается $f(x) = \arg\max_{1\leq j\leq K} \text{sigmoid}\ ||(z(x)) - A_j||_2$.

Несмотря на то, что применять корректирующие коды для задач машинного обучения впервые было предложено 1995 году, исследование их эффективности в противостоянии состязательным атакам остается остро актуальным и в настоящее время.

3.5.2. Разнообразие ансамблей

Ансамбли, в составе которых согласуется работа разнотипных ML-моделей, также является весьма перспективным подходлм в обеспечении защиты от состязательных атак. В статье [53] предлагается ансамбль с дополнительным элементом регуляризации, который поощряет разнообразие моделей. Предполагается, что по сравнению с одиночными моделями или ансамблями с однотипными моделями дополнительное разнообразие моделей затруднит атакам поиск состязательных примеров.

Пусть $f_m(x)$ - вектор вероятности выходов m-й модели в ансамбле, а $f(x) = \Sigma_m f_m(x)$ - вектор вероятности, выводимый ансамблем. В качестве loss-функции ансамбль моделей использует выражение (16):

$$L(x, y) = -\alpha \mathrm{H}\left(f(x)\right) - \beta \mathrm{R}^3\left(\left\{f_m^{\setminus y}(x)\right\}\right) + \sum_{m=1}^{M} L_{CE}\left(f_m(x), y\right), \quad (16)$$

где $\mathrm{H}(\cdot)$ – энтропия Шеннона, L_{CE} – стандартная кросс-энтропия, применяемая к прогнозированию каждой модели, R^3 - гипершар, охватываемый векторами вероятности отдельных (нормализованных) моделей ансамбля. При вычислении R^3 исключают ведущий класс из каждого вектора вероятности $f_m(x)$. Весовые коэффициенты α и β являются гиперпараметрами.

Ни один из членов в функции потерь (16), не поощряет какого-либо рода маскировку градиента. Нет никакого другого механизма, кроме изменения цели обучения, что позволяет предположить атакующему, что стандартные готовые атаки на основе градиента должны быть успешными. В [32] показано, что простое увеличение числа итераций и /или комбинирование PGD с существенно отличающейся атакой, такой как B&B [37], может существенно увеличить успех атаки.

3.5.3. EMPIR

Хорошо известными и успешными с точки зрения повышения качества прогнозирования ML-моделей являются такие методы объединения, как усреднение, пакетирование и бустинг. Механизм защиты EMPIR (Ensembles of Mixed Precision for Increased Robustness) [54] – еще один из механизмов ансамблирования, в котором ансамбль моделей объединяет квантованные глубокие нейросетевые модели со смешанной точностью весов и значений функций активаций. Изначально квантование весов и функций активации использовались для снижения вычислительных затрат при обучении модели. Однако, оказалось, что квантованные модели демонстрируют более высокую устойчивость к атакам противника [54].

В EMPIR обучается несколько моделей f_i с разными уровнями квантования параметров (весов) и/или гиперпараметров (выходов функций активации) для каждой модели, и возвращается результат мажоритарного голосовая моделей. Например, для защиты датасета CIFAR10 в статье [54] используется одна модель с полной точностью, одна модель, обученная с 2-разрядными активациями и 4-разрядными весами, и другая модель с 2-разрядными активациями и 2-разрядными весами.

Данный механизм, по сути, включает оценку градиента обратного прохода в стиле BPDA (маскировку градиента), что является слабым механизмом защиты

В [32] рассмотрена следующая эффективная стратегия адаптивной атаки на данный механизм защиты. Сначала строится функции потерь, которая используется для генерации состязательных примеров. В качестве первоначальной формируется простейшая функция потерь - берутся векторы вероятности классов трех моделей $f_1(x)$, $f_2(x)$, $f_3(x)$ и усредняются по компонентам так, чтобы прогноз агрегированной модели для класса i был равен:

$$\hat{f}(x) = \frac{1}{3}\left(f_1(x) + f_2(x) + f_3(x)\right). \quad (17)$$

Затем выполняется атака PGD для кросс-энтропии модели $\hat{f}(x)$. Используя (17), классификатор принимает мажоритарное голосование для принятия окончательного прогноза. Если два классификатора выносят решение в пользу целевого класса, то третья модель может иметь нулевую достоверность. Выполнив, как показано в [32] 100 итераций PGD для этой функции потерь, атакующий снижает точность модели до 1,5% при ошибке $\varepsilon = 0{,}031$.

При формировании защитных ансамблей следует учитывать.

1) Объединение нескольких ML-моделей в единую комбинированную защищенную модель чрезвычайно эффективно, однако, если найден простой метод атаки на одну из составляющих, он часто оказывается эффективным для атакующего.

2) Существует ряд приемов, каждый из которых может увеличить вероятность успеха атаки на несколько процентных пунктов, когда точность модели уже низкая.

3) Если атака использует случайные начальные точки, атакующий повторит несколько раз атаку для каждого образца.

4) Атакующий не видит смысла использовать похожие атаки (например, BIM, PGD или MIM), и будет стремиться, использовать существенно отличающиеся стратегии атак (например, комплекс из PGD и B&B атак).

5) Ансамбли со слабыми механизмами защиты не обеспечивают эмержентность сильной защиты.

3.6. Стохастическая интерполяция

3.6.1 Mixup Inference

В статье [55] используется процедура стохастической интерполяции во время вывода о пригодности состязательного сигнала. Механизм защиты работает следующим образом. Для каждого входного сигнала x вычисляется K интерполяций с выборками s_k:

$$\tilde{x}_k = \alpha x + (1-\alpha)s_k, \quad (18)$$

где α – фиксированный гиперпараметр (например, $\alpha = 0{,}6$ во всех экспериментах), а s_k выбирается случайным образом из предопределенного набора изображений S. Затем усредняют логит-ответы незащищенной модели по всем K интерполяциям, т.е. окончательный ответ защищенной модели равен:

$$\hat{z}(x) = \frac{1}{K}\sum_{i=1}^{K} z(\tilde{x}_k) = \frac{1}{K}\sum_{i=1}^{K} z(\alpha x - (1-\alpha)s_k), \quad (19)$$

Авторы предлагают несколько способов реализации механизмов защиты при использовании (18) и (19), которые отличаются лишь способом отбора s_k из S. (либо s_k отбирается равномерно из всех изображений, для которых прогнозируемая метка отличается от x, либо s_k отбирается равномерно из всех изображений, для которых предсказанная метка такая же, как и для x. Эвристика авторов базируется на предположениях, что эффект возмущения при атаке уменьшается за счет интерполяции, а также что возмущение имеет тенденцию оказывать влияние на модель только вблизи заданного входного значения x и что большие отклонения ослабляют влияние на модель.

Защита Mixup Inference является особенно интересным примером стохастической защиты, благодаря своему нелокальному механизму микширования, который выполняет интерполяцию между удаленными изображениями. Однако. в [32] показано, что правильно адаптированный алгоритм атаки, учитывающий защитный механизм, способен взломать подобный механизм защиты.

Аналог преобразований Mixup Inference часто применяют при аугментации данных и формировании синтетических датасетов. Особо отметим, что оценку пригодности состязательного сигнала важно выполнять и ходе создания подобных расширенных датасетов, что позволит в последствии не только повысить качество классификаторов, но их устойчивость к состязательным атакам.

При использовании механизмов защиты на базе стохастической интерполяции следует учитывать:

1) При конструировании механизмов защиты следует сосредоточиться на контроле адаптивных атак типа «белый ящик».

2) Если для защиты используются стохастические методы, то при оценке защиты следует стабилизировать градиенты модели путем усреднения по *всем* атакам. Только стабильный градиент позволит атакующему в ходе итеративной атаки на основе градиента добиться цели.

Заключение

Основное внимание при построении механизмов защиты ML-моделей и оценки надежности должно быть сосредоточено на противодействие комплексным адаптивным атакам, которые явно выявляют и нацеливаются на самые слабые звенья защиты. Хотя адаптивные атаки на сегодняшний день дорабатываются вручную для нацеливания на конкретные средства защиты, формируется ряд автоматизированных инструментов (эксплойтов), для всесторонней оценки надежность защиты, которые могут использоваться не всегда легально. Чтобы противостоять этому, сформулируем основные правила, которым нужно следовать при разработке механизмов защиты ML-моделей:

1) Для комплексной защиты требуется проанализировать все этапы функционирования ML-модели, чтобы определить, на каких из них следует сосредоточить внимание с точки зрения защиты

2) Всегда существует адаптивная атака, способная генерировать состязательные образцы, которые так же близки к границам принятия решений и обладают такой же достоверностью, как и доброкачественные примеры.

3) Следует оптимизировать комбинированную функцию потерь: объединение нескольких термов (членов) loss-функции приводит к трудно оптимизируемым функциям потерь, которые могут вести себя не так, как хотелось бы атакующему.

4) Атаки, основанные на деревьях решений, могут работать лучше, чем атаки, основанные на градиентах, в частности, когда градиенты (намеренно) маскируются.

5) Атакующий учитывает случайность на уровне всего механизма защиты. Следовательно, необходимо выполнить оценку защиты от случайного шума

6) Необходимо учесть возможность влияния альтернативных функций потерь (функций-противников).

7) Объединение нескольких ML-моделей в единую комбинированную защищенную модель чрезвычайно эффективно, однако, если найден простой метод атаки на одну из составляющих, он часто оказывается эффективным для атакующего.

8) Ансамбли со слабыми механизмами защиты не обеспечивают сильную защиту

9) Следует учитывать атаки, использующие существенно отличающиеся стратегии, Большая часть оценки защиты должна быть сосредоточена на адаптивных атаках, а не на атаках, которые не обращают внимания на защитный механизм.

Литература

1. *Goldblum M. et al*. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022. Vol. 45. №. 2, pp. 1563-1580.

2. *Zhang J. et al*. Protecting Intellectual Property of Deep Neural Networks with Watermarking // Proceedings of the 2018 on Asia Conference on Computer and Communications Security. 2018, pp 159-172, 2018 DOI:10.1145/3196494.3196550

3. *Adi Y, et al.* Turning your weakness into a strength: Watermarking deep neural networks by backdooring // 27th {USENIX} Security Symposium, 2018, pp. 1615-1631, URL: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-adi.pdf (дата обращения – 05.05.2023)

4. *Fang M., Gong N. Z, and Liu J.* Influence function based data poisoning attacks to top-n recommender systems // Proceedings of The Web Conference 2020, pp. 3019-3025, 2020. DOI:10.1145/3366423.3380072

5. *Fung C., Yoon C., and Beschastnikh I.* Mitigating sybils in federated learning poisoning. URL: https://arxiv.org/pdf/1808.04866.pdf. (дата обращения – 05.05.2023)

6. *Cao D., Chang S., Lin Z., Liu G., Sun D.* Understanding Distributed Poisoning Attack in Federated Learning // 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 2019, pp. 233-239, DOI:10.1109/ICPADS47876.2019.00042

7. *Saha A., Subramanya A., Pirsiavash H.* Hidden trigger backdoor attacks // Proceedings of the AAAI conference on artificial intelligence. 2020. Т. 34. №. 07, pp. 11957-11965.

8. *Huang W. R., et al.* Metapoison: Practical general-purpose clean-label data poisoning. URL: https://doi.org/10.48550/arXiv.2004.00225, 2020 (дата обращения – 05.05.2023)

9. *Gu T., Dolan-Gavitt B., Garg S.* Badnets: Identifying vulnerabilities in the machine learning model supply chain // arXiv preprint arXiv:1708.06733. 2017.

10. *Sun M., Agarwal S., Kolter J. Z.* Poisoned classifiers are not only backdoored, they are fundamentally broken // arXiv preprint arXiv:2010.09080. 2020. URL: https://doi.org/10.48550/arXiv.2010.09080 (дата обращения – 05.05.2023)

11. *Du R. M. et al.* An embarrassingly simple approach for trojan attack in deep neural networks // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp 218-228, URL: https://arxiv.org/pdf/2006.08131.pdf (дата обращения – 05.05.2023)

12. *Liu Y. et al.* Trojaning attack on neural networks. In NDSS, 2018 URL: https://weihang-wang.github.io/papers/tnn_ndss18.pdf (дата обращения – 05.05.2023)

13. *Yao Y., et al.* Latent backdoor attacks on deep neural networks //. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2041-2055, URL: https://dl.acm.org/doi/pdf/10.1145/3319535.3354209 (дата обращения – 05.0 .2023)

14. *Wang Y., Sarkar E., Li W., Maniatakos M. and Jabari S. E.* Stop-and-Go: Exploring Backdoor Attacks on Deep Reinforcement Learning-Based Traffic Congestion Control Systems // IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4772-4787, 2021, DOI:10.1109/TIFS.2021.3114024

15. *Goodfellow I. J., Shlens J., Szegedy C.* Explaining and harnessing adversarial examples // arXiv preprint arXiv:1412.6572. 2014. URL: https://doi.org/10.48550/arXiv.1412.6572 (дата обращения – 05.05.2023)

16. *Bagdasaryan E., Veit A., Hua Y., Estrin D., and Shmatikov V.* How to backdoor federated learning // International Conference on Artificial Intelligence and Statistics, 2020, pp. 2938-2948. PMLR, DOI:10.48550/arXiv.2303.03320

17. *Baruch G., Baruch M., and Goldberg Y.* A little is enough: Circumventing defenses for distributed learning // Advances in Neural Information Processing Systems, 2019, pp. 8635–8645, URL: https://arxiv.org/pdf/1902.06156.pdf (дата обращения – 05.05.2023)

18. *Aghakhani H. et al.* Bullseye polytope: A scalable clean-label poisoning attack with improved transferability // 2021 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2021, pp. 159-178. DOI: 10.1109/EuroSP51992.2021.00021.

19. *Solans D., Biggio B., Castillo C.* Poisoning attacks on algorithmic fairness //Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I. Cham: Springer International Publishing, 2021, pp. 162-177.

20. *Sun G. et al*. Data poisoning attacks on federated machine learning // IEEE Internet of Things Journal. 2021. Vol. 9. № 13, pp. 11365-11375.

21. *Geiping J. et al.* Witches' brew: Industrial scale data poisoning via gradient matching // arXiv preprint arXiv:2009.02276. 2020. URL: https://doi.org/10.48550/arXiv.2009.02276 (дата обращения – 05.05.2023)

22. *Verma V. et al.* Manifold mixup: Better representations by interpolating hidden states //International conference on machine learning. PMLR, 2019, pp. 6438-6447.URL: https://doi.org/10.48550/arXiv.1806.05236 (дата обращения – 05.05.2023)

23. *Ma Y., Zhu X., Hsu J.* Data poisoning against differentially-private learners: Attacks and defenses // arXiv preprint arXiv:1903.09860. 2019. URL: https://doi.org/10.48550/arXiv.1903.09860 (дата обращения – 05.05.2023)

24. *Mahloujifar S., Mahmoody M., Mohammed A.* Universal multi-party poisoning attacks // International Conference on Machine Learning. PMLR, 2019, pp. 4274-4283. URL: https://doi.org/10.48550/arXiv.1809.03474 (дата обращения – 05.05.2023)

25. *Mahloujifar S., Diochnos D. I., Mahmoody M.* The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure // Proceedings of the AAAI Conference on Artificial Intelligence. 2019. Vol. 33. №. 01, pp. 4536-4543. URL: https://doi.org/10.48550/arXiv.1809.03063 (дата обращения – 05.05.2023)

26. *Shen J., Zhu X. and Ma D,* TensorClog: An Imperceptible Poisoning Attack on Deep Neural Network Applications // IEEE Access, 2019. Vol. 7, pp. 41498-41506, , DOI: 10.1109/ACCESS.2019.2905915.

27. *Szegedy C. et al.* Intriguing properties of neural networks //arXiv preprint arXiv:1312.6199. 2013. URL: https://doi.org/10.48550/arXiv.1312.6199 (дата обращения – 05.05.2023)

28. *Sen S., Ravindran B., Raghunathan A.* Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks // arXiv preprint arXiv:2004.10162. 2020. URL: https://doi.org/10.48550/arXiv.2004.10162 (дата обращения – 05.05.2023)

29. *Santurkar S. et al.* Image synthesis with a single (robust) classifier // Advances in Neural Information Processing Systems. 2019. Vol. 32. URL: https://doi.org/10.48550/arXiv.1906.09453 (дата обращения – 05.05.2023)

30. *Athalye, A., Carlini, N., and Wagner, D.* Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples // International Conference on Machine Learning, 2018. URL: https://arxiv.org/pdf/1802.00420.pdf (дата обращения – 05.05.2023)

31. *Carlini N., Wagner D.* Adversarial examples are not easily detected: Bypassing ten detection methods // Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017, pp. 3-14 URL: https://doi.org/10.48550/arXiv.1705.07263 (дата обращения – 05.05.2023)

32. *Tramer F. et al.* On adaptive attacks to adversarial example defenses // Advances in neural information processing systems. 2020. Vol. 33, pp. 1633-1645. URL: https://doi.org/10.48550/arXiv.2002.08347 (дата обращения – 05.05.2023)

33. *Madry A. et al.* Towards deep learning models resistant to adversarial attacks // arXiv preprint arXiv:1706.06083. 2017. URL: https://doi.org/10.48550/arXiv.1706.06083 (дата обращения – 05.05.2023)

34. *Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K.* Synthesizing robust adversarial examples // International Conference on Machine Learning, 2018. URL: https://arxiv.org/pdf/1707.07397.pdf (дата обращения – 05.05.2023)

35. *Gowal S. et al.* An alternative surrogate loss for pgd-based adversarial testing //arXiv preprint arXiv:1910.09338. 2019. URL: https://doi.org/10.48550/arXiv.1910.09338 (дата обращения – 05.05.2023)

36. *lyas A. et al.* Black-box adversarial attacks with limited queries and information //International conference on machine learning. PMLR, 2018, pp. 2137-2146. URL: https://doi.org/10.48550/arXiv.1804.08598 (дата обращения – 05.05.2023)

37. *Brendel W., Rauber J., Bethge M.* Decision-based adversarial attacks: Reliable attacks against black-box machine learning models // arXiv preprint arXiv:1712.04248. 2017. URL: https://doi.org/10.48550/arXiv.1712.04248 (дата обращения – 05.05.2023)

38. *Roth K., Kilcher Y., Hofmann T.* The odds are odd: A statistical test for detecting adversarial examples // International Conference on Machine Learning. PMLR, 2019, pp. 5498-5507, URL: https://doi.org/10.48550/arXiv.1902.04818 (дата обращения – 05.05.2023)

39. *Hu S. et al.* A new defense against adversarial images: Turning a weakness into a strength //Advances in Neural Information Processing Systems. 2019. Vol. 32. URL https://doi.org/10.48550/arXiv.1910.07629 data access – 05.05.2023) (дата обращения – 05.05.2023)

40. *Xiao C., Zhong P., Zheng C.* Resisting adversarial attacks by k-winners-take-all // International Conference on Learning Representations, 2020 URL: https://arxiv.org/pdf/1905.10510v1.pdf (дата обращения – 05.05.2023)

41. *Фомичева С.Г.* Теоретические аспекты квантования баз знаний в мультиагентных системах // Информационно-управляющие системы. 2017. № 3 (88). С. 2-10. DOI: 10.15217/issnl684-8853.2017.3.2

42. *Fomicheva S., Bezzateev S.* Modification of the Berlekamp-Massey algorithm for explicable knowledge extraction by SIEM-agents // Journal of Physics: Conference Series, 2022, 2373(5), 052033. DOI:10.1088/1742-6596/2373/5/052033

43. *Bezzateev S., Fomicheva S.* Soft multi-factor authentication // 2020 Wave Electronics and its Application in Information and Telecommunication Systems, WECONF 2020. 2020. P. 9131537. DOI: 10.1109/WECONF48837.2020.9131537

44. *Беззатеев С.В., Фомичева С.Г., Супрун А.Ф.* Повышение эффективности мультиагентных систем информационной безопасности методами постквантовой криптографии // Проблемы информационной безопасности. Компьютерные системы. 2022. № 4 (52). С. 71-88. DOI: 10.48612/jisp/75dp-p3ed-hrmk

45. *Li Y., Bradshaw J., Sharma Y.* Are generative classifiers more robust to adversarial attacks? // International Conference on Machine Learning, 2019/ URL: https://arxiv.org/pdf/1802.06552.pdf (дата обращения – 05.05.2023)

46. *Yin X., Kolouri S., Rohde G. K.* Gat: Generative adversarial training for adversarial example detection and robust classification //arXiv preprint arXiv:1905.11475. 2019. URL: https://doi.org/10.48550/arXiv.1905.11475 (дата обращения – 05.05.2023)

47. *Croce F., Hein M.* Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In ICML, 2020. URL: https://arxiv.org/pdf/2003.01690.pdf (дата обращения – 05.05.2023)

48. *Carlini N. et al.* On evaluating adversarial robustness // arXiv preprint arXiv:1902.06705. 2019. URL: https://doi.org/10.48550/arXiv.1902.06705 (дата обращения – 05.05.2023)

49. *Bafna M., Murtagh J., Vyas N.* Thwarting adversarial examples: An l0-robust sparse fourier transform. In Advances in Neural Information Processing Systems, pp. 10096-10106, 2018 https://arxiv.org/pdf/1812.05013.pdf (дата обращения – 05.05.2023)

50. *Yang Y. et al*. Me-net: Towards effective adversarial robustness with matrix estimation // arXiv preprint arXiv:1905.11971. 2019. URL: https://doi.org/10.48550/arXiv.1905.11971 (дата обращения – 05.05.2023)

51. *Pang T. et al.* Rethinking softmax cross-entropy loss for adversarial robustness // arXiv preprint arXiv:1905.10626. 2019. URL: https://doi.org/10.48550/arXiv.1905.10626 (дата обращения –05.05.2023)

52. *Verma G., Swami A.* Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks // Advances in Neural Information Processing Systems, 2019, pp. 8643-8653, URL: https://proceedings.neurips.cc/paper/2019/file/cd61a580392a70389e27b0bc2b439f49-Paper.pdf (дата обращения – 05.05.2023)

53. *Pang T. et al.* Improving adversarial robustness via promoting ensemble diversity // International Conference on Machine Learning. PMLR, 2019, pp. 4970-4979. URL: https://doi.org/10.48550/arXiv.1901.08846 (дата обращения – 05.05.2023)

54. *Sen S., Ravindran B., Raghunathan A.* Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks //arXiv preprint arXiv:2004.10162. 2020. URL: https://doi.org/10.48550/arXiv.2004.10162 (дата обращения – 05.05.2023)

55. *Pang T., Xu K., Zhu J.* Mixup inference: Better exploiting mixup to defend adversarial attacks // arXiv preprint arXiv:1909.11515. 2019. URL: https://doi.org/10.48550/arXiv.1909.11515 (дата обращения – 05.05.2023)

THE PASSIVE RADIO-FREQUENCY TAGS USAGE IN DECENTRALIZED INFORMATION EXCHANGE SYSTEMS

Svetlana G. Fomicheva, *Petersburg University of Aerospace Instrumentations, Saint Petersburg, Russia,* ***levikha@mail.ru***
Sergey V. Bezzateev, *Petersburg University of Aerospace Instrumentations, Saint-Petersburg, Russia,* ***bsv@aanet.ru***

Abstract
The problems of protection for intelligent information systems are acutely relevant due to their application in the subjects of critical information infrastructure. The most difficult to identify are adversarial attacks on machine learning models, which are carried out during transfer and federative training of already pre-trained models. At the same time, the anatomy of adversarial attacks is rarely covered in the Russian-language segment of publications, and the defenses against them and mechanisms for evaluating protection against attacks on machine learning models are practically absent, which actualizes the need for the analytical review presented in the article. Purpose: the purpose of the study is to conduct an analytical review and a formalized description of the defenses for machine learning models that are the target of adversarial attacks. Results: based on the classification of attacks aimed at machine learning models, the modern principles of their defenses are formalized. Unlike existing publications, our review highlights and summarizes not only the types of attacks on machine learning models, but also the mechanisms of their implementation. The classification of existing methods of protection against adversarial attacks is carried out. Practical relevance: as a result of generalization, the necessary requirements for the construction of models protected from adversarial attacks are formulated. Discussion: The main attention when building protection mechanisms for machine learning models and evaluating their reliability should be focused on countering complex adaptive attacks that clearly identify and target the weakest links of protection.

Keywords: *machine learning models; adversarial attack; defense mechanisms; loss function optimization; gradient masking; model ensembles.*

References

1. M. Goldblum et al., "Dataset security for machine learning: Data poisoning, backdoor attacks, and defensesm" *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2022. Vol. 45. No. 2, pp. 1563-1580.
2. J. Zhang et al., "Protecting Intellectual Property of Deep Neural Networks with Watermarking," *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 2018, pp 159-172, 2018 DOI:10.1145/3196494.3196550
3. Y. Adi, et al., "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," *27th {USENIX} Security Symposium*, 2018, pp.1615-1631, URL: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-adi.pdf (data access - 05.05.2023)
4. M. Fang, N.Z. Gong, and J. Liu, "Influence function based data poisoning attacks to top-n recommender systems," *Proceedings of The Web Conference* 2020, pp. 3019-3025. DOI:10.1145/3366423.3380072
5. C. Fung, C. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," URL: https://arxiv.org/pdf/1808.04866.pdf. (data access - 05.05.2023)
6. D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding Distributed Poisoning Attack in Federated Learning," *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS),* Tianjin, China, 2019. Pp. 233-239, DOI:10.1109/ICPADS47876.2019.00042
7. A. Saha, A. Subramanya, H. Pirsiavash, "Hidden trigger backdoor attacks," *Proceedings of the AAAI conference on artificial intelligence*. 2020. Vol. 34. No. 07, pp. 11957-11965.
8. W.R. Huang, et al., "Metapoison: Practical general-purpose clean-label data poisoning," 2020. URL: https://doi.org/10.48550/arXiv.2004.00225, (data access - 05.05.2023)
9. T. Gu, B. Dolan-Gavitt, S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," arXiv preprint arXiv:1708.06733. 2017.
10. M. Sun, S. Agarwal, J.Z. Kolter, "Poisoned classifiers are not only backdoored, they are fundamentally broken," arXiv preprint arXiv:2010.09080. 2020. URL: https://doi.org/10.48550/arXiv.2010.09080 (data access - 05.05.2023)
11. R.M. Du et al.. "An embarrassingly simple approach for trojan attack in deep neural networks," P*roceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*," 2020, pp. 218-228, URL: https://arxiv.org/pdf/2006.08131.pdf (data access - 05.05.2023)
12. Y. Liu et al., "Trojaning attack on neural networks," *NDSS*, 2018 URL: https://weihang-wang.github.io/papers/tnn_ndss18.pdf (data access - 05.05.2023)
13. Y. Yao, et al., "Latent backdoor attacks on deep neural networks," *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2041-2055, URL: https://dl.acm.org/doi/pdf/10.1145/3319535.3354209 (data access - 05.05.2023)
14. Y. Wang, E. Sarkar, W. Li, M. Maniatakos, and S.E. Jabari, "Stop-and-Go: Exploring Backdoor Attacks on Deep Reinforcement Learning-Based Traffic Congestion Control Systems," *IEEE Transactions on Information Forensics and Security*, 2021. Vol. 1, pp. 4772-4787. DOI:10.1109/TIFS.2021.3114024
15. I.J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572. 2014. URL: https://doi.org/10.48550/arXiv.1412.6572 (data access - 05.05.2023)
16. E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics," 2020, pp. 2938-2948. PMLR, DOI:10.48550/arXiv.2303.03320
17. G. Baruch, M. Baruch, Y. and Goldberg, "A little is enough: Circumventing defenses for distributed learning," Advances in Neural Information Processing Systems, 2019. Pp. 8635-8645, URL: https://arxiv.org/pdf/1902.06156.pdf (data access - 05.05.2023)
18. H. Aghakhani, et al., "Bullseye polytope: A scalable clean-label poisoning attack with improved transferability," *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2021, pp. 159-178. DOI: 10.1109/EuroSP51992.2021.00021.
19. D. Solans, B. Biggio, C. Castillo, "Poisoning attacks on algorithmic fairness.Machine Learning and Knowledge Discovery in Databases: European Conference," *ECML PKDD 2020*, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I. Cham : Springer International Publishing, 2021, pp. 162-177.
20. G. Sun et al., "Data poisoning attacks on federated machine learning," *IEEE Internet of Things Journal.* 2021. Vol. 9. No. 13, pp. 11365-11375.
21. J. Geiping et al., "Witches' brew: Industrial scale data poisoning via gradient matching," arXiv preprint arXiv:2009.02276. 2020. URL: https://doi.org/10.48550/arXiv.2009.02276 (data access - 05.05.2023)
22. V. Verma et al., "Manifold mixup: Better representations by interpolating hidden states. International conference on machine learning," *PMLR*, 2019, pp. 6438-6447.URL: https://doi.org/10.48550/arXiv.1806.05236 (data access - 05.05.2023)
23. Y. Ma, X. Zhu, J. Hsu, "Data poisoning against differentially-private learners: Attacks and defenses," arXiv preprint arXiv:1903.09860. 2019. URL: https://doi.org/10.48550/arXiv.1903.09860 (data access - 05.05.2023)
24. S. Mahloujifar, M. Mahmoody, A. Mohammed, "Universal multi-party poisoning attacks International Conference on Machine Learning," *PMLR*, 2019, pp. 4274-4283. URL: https://doi.org/10.48550/arXiv.1809.03474 (data access - 05.05.2023)

25. S. Mahloujifar, D.I. Diochnos, M. Mahmoody, "The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure," *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019. Vol. 33. No. 01, pp. 4536-4543. URL: https://doi.org/10.48550/arXiv.1809.03063 (data access - 05.05.2023)
26. Shen J., Zhu X. and Ma D, TensorClog: An Imperceptible Poisoning Attack on Deep Neural Network Applications, In IEEE Access, 2019. Vol. 7, pp. 41498-41506, DOI: 10.1109/ACCESS.2019.2905915.
27. C. Szegedy, et al., "Intriguing properties of neural network.s/arXiv preprint arXiv:1312.6199," 2013. URL: https://doi.org/10.48550/arXiv.1312.6199 (data access - 05.05.2023)
28. S. Sen, B. Ravindran, A. Raghunathan, "Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks," arXiv preprint arXiv:2004.10162. 2020. URL: https://doi.org/10.48550/arXiv.2004.10162 (data access - 05.05.2023)
29. S. Santurkar S. et al., "Image synthesis with a single (robust) classifier," *Advances in Neural Information Processing Systems*. 2019. Vol. 32. URL: https://doi.org/10.48550/arXiv.1906.09453 (data access - 05.05.2023)
30. A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *International Conference on Machine Learning*, 2018. URL: https://arxiv.org/pdf/1802.00420.pdf (data access - 05.05.2023)
31. N. Carlini, D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 2017, pp. 3-14 URL: https://doi.org/10.48550/arXiv.1705.07263 (data access - 05.05.2023)
32. F. Tramer et al., "On adaptive attacks to adversarial example defenses," *Advances in neural information processing systems*. 2020. Vol. 33, pp. 1633-1645. URL: https://doi.org/10.48550/arXiv.2002.08347 (data access - 05.05.2023)
33. A. Madry et al., "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083. 2017. URL: https://doi.org/10.48550/arXiv.1706.06083 (data access - 05.05.2023)
34. A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," *International Conference on Machine Learning*, 2018. URL: https://arxiv.org/pdf/1707.07397.pdf (data access - 05.05.2023)
35. S. Gowal et al., "An alternative surrogate loss for pgd-based adversarial," arXiv preprint arXiv:1910.09338. 2019. URL: https://doi.org/10.48550/arXiv.1910.09338 (data access - 05.05.2023)
36. A. Iyas et al., "Black-box adversarial attacks with limited queries and information," *International conference on machine learning. PMLR*, 2018, pp. 2137-2146. URL: https://doi.org/10.48550/arXiv.1804.08598 (data access - 05.05.2023)
37. W. Brendel, J. Rauber, M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," arXiv preprint arXiv:1712.04248. 2017. URL: https://doi.org/10.48550/arXiv.1712.04248 (data access - 05.05.2023)
38. K. Roth, Y. Kilcher, T. Hofmann, "The odds are odd: A statistical test for detecting adversarial examples," *International Conference on Machine Learning. PMLR*, 2019, pp. 5498-5507, URL: https://doi.org/10.48550/arXiv.1902.04818 (data access - 05.05.2023)
39. S. Hu et al., "A new defense against adversarial images: Turning a weakness into a strength," *Advances in Neural Information Processing Systems*. 2019. Vol. 32. URL https://doi.org/10.48550/arXiv.1910.07629 data access - 05.05.2023) (data access - 05.05.2023)
40. C. Xiao, P. Zhong, and C. Zheng, "Resisting adversarial attacks by k-winners-take-all," *International Conference on Learning Representations*, 2020 URL: https://arxiv.org/pdf/1905.10510v1.pdf (data access - 05.05.2023)
41. S.G. Fomicheva, "Theoretical aspects of knowledge base quantization in multi-agent systems," Information and control systems. 2017. No.3 (88), pp. 2-10. DOI: 10.15217/issnl684-8853.2017.3.2 (In Russian)
42. S. Fomicheva, and S. Bezzateev, "Modification of the Berlekamp-Massey algorithm for explicable knowledge extraction by SIEM-agents," *Journal of Physics: Conference Series*, 2022, no. 2373(5), 052033. DOI:10.1088/1742-6596/2373/5/052033
43. S. Bezzateev, S. Fomicheva, "Soft multi-factor authentication," *Proceeding: 2020 Wave Electronics and its Application in Information and Telecommunication Systems, WECONF*. 2020. P. 9131537. DOI: 10.1109/WECONF48837.2020.9131537
44. S.V. Bezzateev, S.G. Fomicheva, A.F. Suprun, "Improving the efficiency of multi-agent information security systems using post-quantum cryptography," *Problems of information security. Computer systems*, 2022. No.4 (52), pp. 71-88. DOI: 10.48612/jisp/75dp-p3ed-hrmk (In Russian)
45. Y. Li, J., Bradshaw, and Y. Sharma, "Are generative classifiers more robust to adversarial attacks?" *International Conference on Machine Learning*, 2019 URL: https://arxiv.org/pdf/1802.06552.pdf (data access - 05.05.2023)
46. X. Yin, S. Kolouri, G.K. Rohde, "Gat: Generative adversarial training for adversarial example detection and robust classification," arXiv preprint arXiv:1905.11475. 2019. URL: https://doi.org/10.48550/arXiv.1905.11475 (data access - 05.05.2023)
47. F. Croce, and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," *ICML*, 2020. URL: https://arxiv.org/pdf/2003.01690.pdf (data access - 05.05.2023)
48. N. Carlini et al., "On evaluating adversarial robustness," arXiv preprint arXiv:1902.06705. 2019. URL: https://doi.org/10.48550/arXiv.1902.06705 (data access - 05.05.2023)
49. M. Bafna, J. Murtagh, and N. Vyas, "Thwarting adversarial examples: An l0-robust sparse fourier transform," *Advances in Neural Information Processing Systems*, pp. 10096-10106, 2018 https://arxiv.org/pdf/1812.05013.pdf (data access - 05.05.2023)
50. Y. Yang et al., "Me-net: Towards effective adversarial robustness with matrix estimation," arXiv preprint arXiv:1905.11971. 2019. URL: https://doi.org/10.48550/arXiv.1905.11971 (data access - 05.05.2023)
51. T. Pang et al., "Rethinking softmax cross-entropy loss for adversarial robustness," arXiv preprint arXiv:1905.10626. 2019. URL: https://doi.org/10.48550/arXiv.1905.10626 (data access - 05.05.2023)
52. G. Verma, and A. Swami, "Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks," *Advances in Neural Information Processing Systems*, 2019, pp. 8643-8653, URL: https://proceedings.neurips.cc/paper/2019/file/cd61a580392a70389e27b0bc2b439f49-Paper.pdf (data access - 05.05.2023)
53. T. Pang et al., "Improving adversarial robustness via promoting ensemble diversity," *International Conference on Machine Learning. PMLR*, 2019, pp. 4970-4979. URL: https://doi.org/10.48550/arXiv.1901.08846 (data access - 05.05.2023)
54. S. Sen, B. Ravindran, A. Raghunathan, "Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks," arXiv preprint arXiv:2004.10162. 2020. URL: https://doi.org/10.48550/arXiv.2004.10162 (data access - 05.05.2023)
55. T. Pang, K. Xu, J. Zhu, "Mixup inference: Better exploiting mixup to defend adversarial attacks," arXiv preprint arXiv:1909.11515. 2019. URL: https://doi.org/10.48550/arXiv.1909.11515 (data access - 05.05.2023)

Information about authors

Svetlana G. Fomicheva, PhD, Full Professor, Professor at the Department of Information Security, St. Petersburg University of Aerospace Instrumentations, Saint-Petersburg, Russia

Sergey V. Bezzateev, Ph.D., Associate Professor, Head of Information Security Department, Saint-Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, Russia