Схемы атак на модели машинного обучения

Д.Е. Намиот

***Аннотация*—В настоящей статье рассматриваются схемы атак на системы искусственного интеллекта (на модели машинного обучения). Классически, атаки на системы машинного обучения - это специальные модификации данных на одном из этапов конвейера машинного обучения, которые призваны воздействовать на модель необходимым атакующему образом. Атаки могут быть направлены на то, чтобы понизить общую точность или честность модели, или же на то, чтобы, например, обеспечить, при определенных условиях, необходимый результат классификации. Другие формы атак могут включать непосредственное воздействие на модели машинного обучения (их код) с теми же целями, что и выше. Есть еще специальный класс атак, который направлен на извлечение из модели ее логики (алгоритма) или информации о тренировочном наборе данных. В последнем случае не происходит модификации данных, но используются специальным образом подготовленные множественные запросы к модели.**

Общей проблемой для атак на модели машинного обучения является тот факт, что модифицированные данные есть такие же легитимные данные, что и не подвергшиеся модификации. Соответственно нет явного способа однозначно определить такого рода атаки. Их эффект в виде неправильного функционирования модели может проявиться и без целенаправленного воздействия. По факту, атакам подвержены все дискриминантные модели.

***Ключевые слова*—машинное обучение, кибератаки, кибербезопасность ИИ**

I. Введение

Эта статья является продолжением серии публикаций, посвященных атакам на системы машинного обучения [1, 2]. 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" [3]. В статье излагаются материалы, которые были представлены в лекциях “Введение в атаки на системы машинного обучения” и “Схемы атак на системы машинного обучения”. Также эти материалы представляются в рамках спецкурса Департамента Кибербезопасности Сбербанка в рамках магистерской программы ПОВС [4].

Статья получена 14 марта 2023. Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета «Мозг, когнитивные системы, искусственный интеллект»

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

Системы машинного обучения (а, по крайней мере, сейчас – это синоним для систем искусственного интеллекта) зависят от данных. Это тавтологическое утверждение приводит, на самом деле, к достаточно серьезным последствиям. Изменение данных тогда, вообще говоря, изменяет работу модели. Но, модели машинного обучения всегда обучаются на некотором подмножестве данных (тренировочном наборе). А уже затем, на этапе эксплуатации модель встречается с реальными данными из генеральной совокупности. И вот эти данные могут по своим характеристикам, вообще говоря, отличатся от тех, на которых модель обучалась и тестировалась. На рисунке 1 представлена типичная ситуация с моделью машинного обучения.

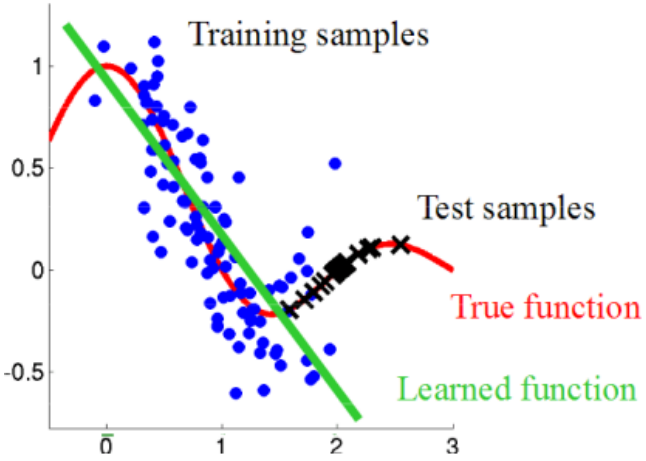

Рис. 1. Аппроксимация неизвестной функции [5]

Мы пытаемся предсказать аппроксимировать неизвестную зависимость, представленную красной линией. Синим обозначены данные тренировочного (тестового) набора (рис. 2).

Они прекрасно обобщаются, и прямая зеленая линия соответствует хорошим метрикам модели до тех пор, пока реальные данные соответствуют по характеристикам тренировочному набору. Если данные изменятся (черный цвет – это нормальное распределение, как и в тренировочных данных, но среднее значение уже другое) – потери растут. Изменение данных приводит к изменению (нарушению) работы модели. При этом, эти измененные данные являются такими же легитимными данными для модели, как и те, на которых модель тренировалась (тестировалась).

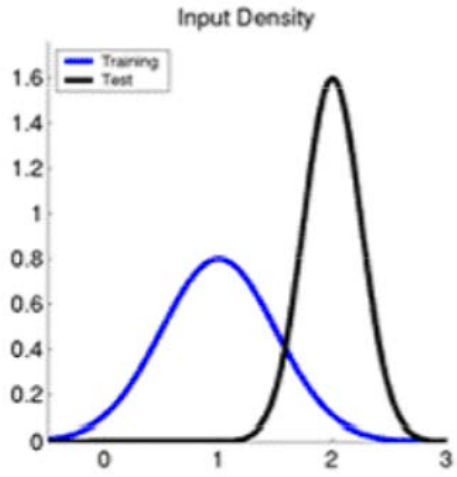

Рис.2. Тренировочные и реальные данные [5]

То есть, самым обычным (естественным) образом, модель машинного обучения при работе на реальных данных может не показывать метрик, которые были достигнуты при тренировке модели и подтверждены при тестировании. При этом на рисунках 1-2 представлена самая простая форма отличия реальных данных от тренировочного набора – так называемый ковариантный (ковариционный) сдвиг. Между тем самые большие проблемы вызывает так называемый сдвиг концепции - изменение связей между входными и выходными переменными. Классический пример последнего времени: построили модель посещения кафе по историческим данным, а COVID изменил поведение посетителей. Старые шаблоны больше не работают. Отсюда, кстати, следует, что мониторинг входных данных в промышленных применениях машинного обучения является обязательным [6].

А что если для используемой модели найдется противоборствующая сторона, которая целенаправленно будет искать такие влияющие на работу модели входные данные? Особенно остро этот вопрос может встать для критических применений (авионика, автоматическое вождение, кибербезопасность и т.д.). Каким образом можно это сделать, например? Разница между прямой и реальной функцией на рисунке 1 и есть потери. Поведение функции потерь определяется стандартно – ее производной по входу. В случае множества характеристик будут, соответственно, частные производные и их совокупность - якобиан. Вот на рисунке 3 и представлена первая работа по так называемым атакам уклонения (специальным модификациям входных данных), которые меняют результат классификации после добавления к правильно распознаваемому изображению шума, вычисленного на основании якобиана функции потерь.

Можно заметить, что такие модификации входных данных тесно связаны с проблемой устойчивости моделей машинного обучения. Классически, алгоритм, в котором погрешность, допущенная в начальных данных или допускаемая при вычислениях, с каждым шагом не увеличивается или увеличивается незначительно, называется устойчивым. В противном случае, если погрешность существенно увеличивается от шага к шагу, алгоритм называется неустойчивым. Устойчивость алгоритма – это мера его чувствительности к изменениям в исходных данных.

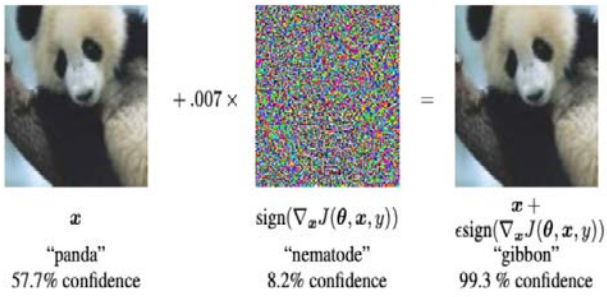

Рис.3. Добавление “шума” обманывает классификатор [7]

Под устойчивым (надежным) машинным обучением обычно понимается устойчивость (надежность) алгоритмов машинного обучения. Чтобы алгоритм машинного обучения считался надежным, ошибка на этапе тестирования (эксплуатации) должна согласовываться с ошибкой обучения. Это означает, что производительность (качество) работы системы остается стабильной на новых данных [8].

Формально, это определяется обычно так: для входных данных *x* и модели *f*, мы хотим, чтобы предсказания модели (например, классификация) оставались такими же для входных данных *x’* в окрестности *x*, где эта окрестность определяется некоторой функцией измерения расстояния δ и максимальной дистанцией Δ:

$$\forall x', \delta(x, x') \leq \Delta \Rightarrow f(x) = f(x') \qquad (1)$$

Например, некоторый классификатор C is δ-стабилен в точке $\vec{X}$ только и если

$$\left\| \vec{X} - \vec{X}_0 \right\|_\infty \leq \delta \Rightarrow C(\vec{X}) = C(\vec{X}_0) \qquad (2)$$

Если мы рассмотрим область, выделенную желтым на фрагменте рисунка 1, то можно заметить, что малые изменения аргумента (увеличение по оси X) вызывает большое увеличение потерь (рисунок 4). То есть мы подобрали данные, при которых модель ведет себя неустойчиво.

Именно устойчивость играет критическую роль в плане использования моделей машинного обучения в критических применениях. Очевидно, что классическое определение устойчивости при этом должно быть расширено. Для критических применений, модель машинного обучения должна сохранять полученные на этапе тестирования метрики на всей генеральной совокупности данных [9].

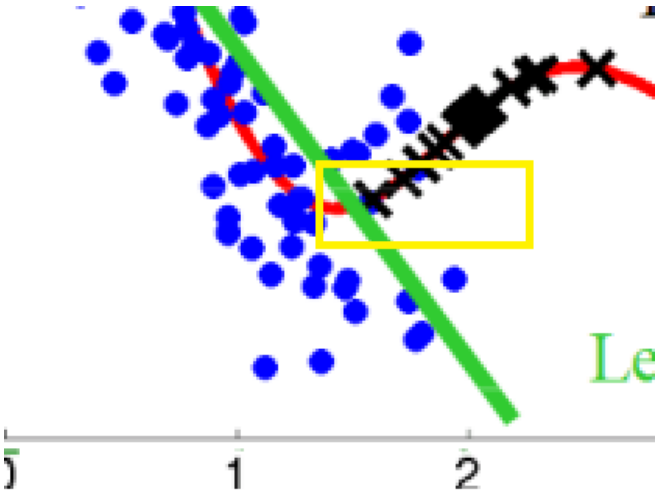

Рис. 4. Нарушение устойчивости с ростом аргумента

Другие примеры изменения работы модели из-за изменения входных данных приведены ниже. Изменение точки зрения (взгляд со стороны и от первого лица - рис. 5) полностью меняет распознавание.

Рис. 5. Изменение точки зрения меняет распознавание [10].

Изменение физических условий (погоды, например) изменяет решение автопилота (рис. 6).

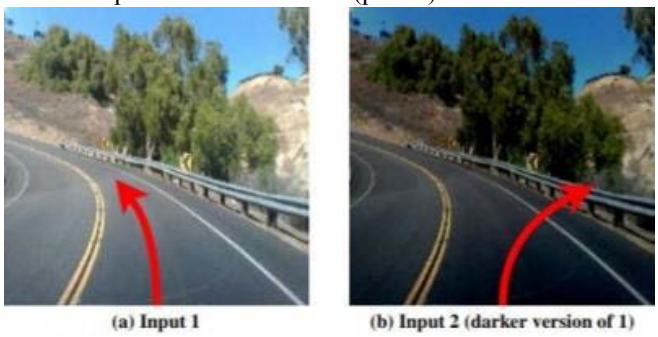

Рис.6. Более темная картинка меняет решение автопилота [11]

Однако модификациями входных данных проблемы моделей машинного обучения не ограничиваются. Данные можно менять специальным образом и на этапе тренировки. Собственно говоря, данные на этапе тренировки модели непосредственно эту модель и определяют. Соответственно, модификации тренировочных данных изменят (могут изменить) модель по сравнению с оригинальными данными. Для двух наборов данных на рисунке 7 обобщения в результате обучения будут разными (угол наклона регрессионной прямой или гиперплоскости SVM разный).

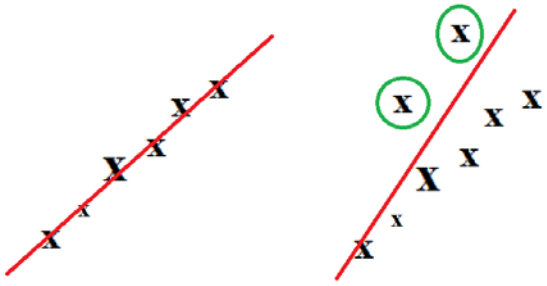

Рис. 7. Зависимость модели от тренировочных данных [12]

Вопрос: обведенные зеленым две точки в правом наборе данных – это реальные данные, аномалии (ошибка измерения) или сознательно помещенные в тренировочный набор данные, призванные изменить работу модели?

Как такие специальные данные могли попасть в тренировочный набор (если исключить умысел разработчиков)? На самом деле, тут все достаточно просто. Данные для тренировки могли быть загружены из какого-то публичного репозитория, куда их разместил атакующий. Это наиболее реальный путь – вспомните, когда вы последний раз занимались анализом загруженных датасетов на предмет наличия отравленных данных?

Другой, также абсолютно реальный путь - это оутсорсинг. Нужные атакующему данные могли возникнуть во время проведения “правильной” разметки.

Атаки на системы машинного обучения – это сознательная (специальная) модификация данных на различных этапах конвейера машинного обучения, которая призванная либо снизить общее качество работы системы (вплоть до полной неработоспособности), либо добиться желаемого функционала (показателей работы). Если специальных целей по тому, как должна работать атакованная система нет, то говорят о нецелевых атаках, в противном случае – атаки целевые.

Модифицированные данные остаются такими же легитимными данными, как и все остальные. В этом и заключается одна из основных проблем с атаками на модели машинного обучения – нет явного способа сказать, что это именно сознательная модификация данных, а не ошибка, аномалия или даже нормальное поведение, которое не отмечалось на тренировочном наборе данных, но присутствует в генеральной совокупности.

Помимо модификации данных оказывается возможным атаковать и непосредственно модели (их код) – выявлять скрытую информацию посредством специально организованных запросов или даже

вмешиваться в работу моделей.

В целом, атаки на системы машинного обучения (системы искусственного интеллекта) относятся к одной из областей использования искусственного интеллекта в кибербезопасности – кибербезопасности самих систем искусственного интеллекта [13].

II. Таксономия

Как описывают атаки? Достаточно общей является следующая классификация:

- Место и время применения
- Знания об атакуемой системе
- Цели и задачи атак
- Предмет приложения: цифровые или реальные объекты
- Предметная область (домен)

Атаки могут осуществляться на разных этапах: тренировка системы (атака на алгоритм) или ее использование (атака на модель). Методы осуществления атак могут быть разные. Цели атакующего могут быть разные – шпионаж (кража информации), саботаж (препятствование работе). Все это может приводить к разным атакам.

Атаки могут использовать уклонения, отравления, трояны (бэкдоры), перепрограммирование и извлечение скрытой информации. Уклонение (чаще всего это и называют состязательными атаками) и отравление являются наиболее распространенными в настоящее время.

Классификации бывают разные, но, в целом, в них есть общие элементы. Один из примеров классификации приведен ниже.

Этап конвейера\Цель	Шпионаж	Препятствие работе	Мошенничество
Тренировка	Отравление для последующего раскрытия данных	Отравление, Трояны, бэкдоры	Отравление
Исполнение	Атака для раскрытия тренировочных данных	Перепрограммирование Уклонение (ложноотрицательное)	Уклонение (ложноположительное)

Рис.8. Пример классификации атак

Другой пример (другой набор признаков) – на рисунке 9.

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

Рис.9. Пример классификации атак

Физические и цифровые атаки различаются по месту изменения данных: реальные или цифровые объекты (рис. 10).

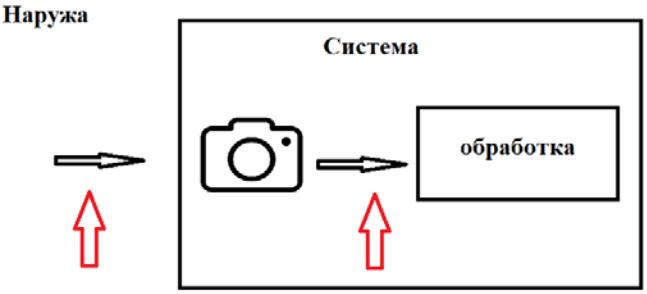

Рис. 10 Физические и цифровые атаки

Атака на рисунке 3 – цифровая. А первая физическая атака получилась, когда модифицированное изображение просто сфотографировали на камеру телефона.

Атаки (вредоносные искажения) могут различаться по своим целям: таргетированные (целевые) атаки, нецелевые или универсальные. Например, если мы хотим изменить результат конкретной классификации – это целевая атака, если хотим вообще ухудшить работу классификатора – нецелевая. Целевые атаки сложнее, чем нецелевые, но полный список может выглядеть так:

1) атакующий пытается оказать воздействие на систему машинного обучения (воспрепятствовать ее правильной работе)
2) атакующий хочет добиться специального результата в работе модели

Или в другой формулировке:

1) атакующий манипулирует данными, чтобы оказать воздействие на систему машинного обучения (воспрепятствовать ее правильной работе)
2) атакующий манипулирует логикой работы, чтобы добиться специального результата в работе.

Если говорить о предметных областях, то тут необходимо отметить следующее. Все дискриминантные модели машинного обучения подвержены атакам. Самая большая проблема, очевидно, это критические приложения (авионика, автоматическое вождение и т.п.). Большинство приложений в таких областях – это классификаторы. Сообразно исследованию проектов в области устойчивых моделей [25], предметные области описываются следующим списком: Данные (для критических приложений):

- изображения (видео) - классификация,
- звук – искажение (изменение) смысла, классификация,
- текст - классификация
- временные ряды – поиск аномалий

Среди моделей отдельно можно выделить атаки на графовые модели машинного обучения.

Говоря о значимости атак можно отметить, что атаки уклонением (модификация входных данных), потенциально, являются наиболее частыми. Если модель требует для работы входные данные, то их можно

пытаться модифицировать нужным образом.

С другой стороны, атаки отравления данных носят долговременный характер. Троян (бэкдор), присутствующий в модели, сохраняется даже после ее переобучения.

В последнее время публикуется все больше работ, которые говорят о важности атак кражи интеллектуальной собственности [26]. Причина состоит в том, что современные модели машинного обучения накапливают множество данных с избыточностью много большей, чем традиционные базы данных. Вот эти избыточные данные и становятся целью атаки. Вместе с тем – извлечение данных (параметров модели etc.) всегда связано с опросами системы. В критических приложениях может просто не быть никакого публичного API. Но он будет в MLaaS (машинное обучение как сервис).

Атаки (методы генерации вредоносных искажений) могут быть разными, в зависимости от имеющихся у атакующего знаний об атакуемой системе. В этой связи говорят о трех типах атак:

- белый ящик,
- черный ящик,
- серый ящик

Белый ящик – полная информация о системе, включая данные обучения. Методы серого ящика - злоумышленник может знать подробную информацию о наборе данных или типе нейронной сети, ее структуре, количестве слоев и т. д. Фактически – нужно делать копию атакуемой модели. Метод черного ящика - злоумышленник может только отправить информацию в систему и получить простой ответ о классе.

В большинстве случаев рассматриваются именно атаки на системы классификации (чаще всего это и есть критические применения). Очевидно, что черный ящик - наиболее реалистичный сценарий.

Теневой моделью называется копия атакуемой системы, на которой можно отрабатывать атаки. Отсюда – информация о параметрах используемых моделей в реальных применениях не должна являться публичной. Модели – публичны, применения – нет. Особенно критично, если атакующий имеет информацию о тренировочных наборах данных. Отсюда – требование к сокрытию информации не только о параметрах модели, но и (особенно) тренировочных наборах данных.

Организация MITRE поддерживает структурированную матрицу используемых киберпреступниками приемов, чтобы облегчить задачу реагирования на киберинциденты [27]. В такой же матрице собирается и информация о состязательных атаках на системы машинного обучения - Adversarial ML Threat Matrix [28].

III. Основания для атак

Почему вообще существуют атаки? На сегодняшний день в сообществе нет единого мнения относительно того, почему это могло быть. Существует ряд объяснений, не всегда согласующихся друг с другом.

Первая и оригинальная гипотеза, пытающаяся объяснить состязательные примеры, была взята из собственной статьи Сегеди [30], где авторы утверждали, что такие примеры существуют из-за наличия маловероятных «карманов» в многообразии (то есть слишком большой нелинейности) и плохой регуляризации сетей. Отсюда, между прочим, оверфиттинг (переобучение) – это проблема с устойчивостью (рис. 10).

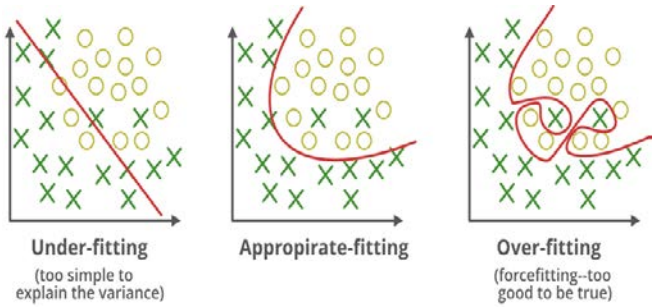

Рис. 10. Under-fitting & over-fitting [29]

Или в более общем плане – все ошибки в машинном обучении связаны именно с устойчивостью [8]. Вот рисунок из работы [32], который иллюстрирует связь состязательных возмущений и устойчивости

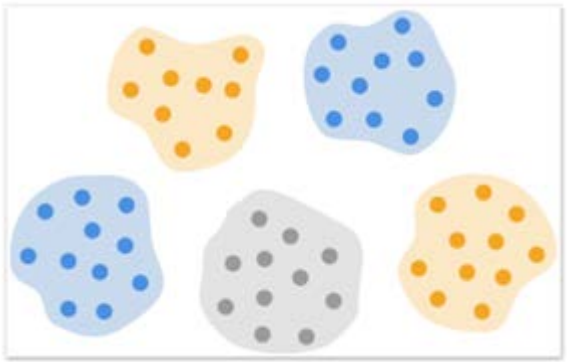

Рис. 11. Устойчивая классификация [32]

Устойчивый классификатор существует, если изменения данных (смещение точек) меньше, чем граница между кластерами.

Противоположная теория, впервые предложенная Гудфеллоу [7], говорит о том, что на самом деле состязательные примеры возникали из-за слишком большой линейности в современном машинном обучении и, особенно, в системах глубокого обучения.

Гудфеллоу утверждал, что функции активации, такие как *ReLU* и *Sigmoid*, в основном представляют собой прямые линии. Итак, на самом деле внутри нейронной сети у вас есть много функций, которые сохраняют вклад друг друга в одном направлении. Если вы затем добавите крошечные возмущения к некоторым входам (несколько пикселей здесь и там), которые накапливаются в огромную разницу на другом конце сети, она выдаст непонятный результат. Вот свежая

работа на эту тему [31], где показывается, что большинство сетей с *ReLU* активацией подвержены атакам с изменением исходных данных по мере ℓ_2.

Третья и, возможно, наиболее часто принимаемая сегодня гипотеза - это объяснение, заключающееся в том, что, поскольку модель никогда не соответствует данным идеально (в противном случае точность набора тестов всегда была бы 100%), всегда будут существовать враждебные карманы входных данных, которые существуют между границей классификатора и фактической подгруппой множества выборочных данных.

Принципиальный момент состоит в том, что при обучении мы используем только какое-то подмножество данных. И, вообще говоря, не знаем всего о генеральной совокупности. Тогда наличие неизвестных системе примеров следует признать свойством выбранного нами подхода, а никак не исключением. Атака – это сознательная генерация (подбор) состязательных примеров, которые существуют (могут существовать) и без злонамеренных пользователей.

Отметим, что в некоторых случаях понимание природы данных (“физики” проблемы) может помочь оценить генеральную совокупность.

IV. Атаки уклонения

Уклонение (часто просто обозначается как состязательные примеры) - наиболее распространенная атака на модель машинного обучения, выполняемая во время вывода (работы модели). Классически, это относится к выработке входных данных, которые кажутся нормальными для человека, но ошибочно классифицируются моделями машинного обучения.

Типичный пример: изменение некоторых пикселей в изображении перед загрузкой, чтобы система распознавания изображений не могла классифицировать результат. Фактически, этот враждебный пример, который может обмануть систему. Идея малозаметности изменений имеет вполне ясную трактовку – если изменения данных незаметны (человек не видит разницы), то и результаты классификации, например, не должны различаться. Это соответствует и приведенному выше определению устойчивости. Но, необходимо отметить, что, вообще говоря, не все системы машинного обучения включают человека в свою работу, и сама идея обмануть человека может оказаться ложной. Для тех же критических применений более характерной будет автоматическая система, которая принимает решения о классификации самостоятельно. В этом случае требований минимального изменения исходных данных, фактически, нет.

Любые допустимые входные данные, которые вызывают неверную работу системы – вредоносные. Отсюда – интерес к семантически обусловленным модификациям (контрфактическим примерам) [33].

На самом деле, о малых изменениях говорят потому, что это позволяет использовать формальные методы в описании и решении. Подробнее об этом рассказано в нашей работе [9].

Атаки «белого ящика» обычно имеют полный доступ к параметрам модели, архитектуре, программе обучения и гипер-параметрам обучения. Часто они являются наиболее мощными атаками, используемыми в литературе. Атаки белого ящика используют информацию о градиенте, чтобы находить состязательные примеры.

Атаки «черного ящика» практически не имеют доступа к параметрам модели, и модель абстрагируется как своего рода API. Атаки черного ящика могут быть запущены с использованием методов оптимизации без градиента, таких как:

- генетические алгоритмы,
- случайный поиск и
- стратегии эволюции.

Обычно они не очень эффективны с точки зрения вычислительных ресурсов, но представляют собой наиболее реалистичный класс атак.

Формально, состязательные модификации описываются следующим образом. Пусть f – наша модель, x – вход, для которого предсказывается выход y. Тогда, состязательный пример d для модели f и входа x может быть определен как:

$f(x+d) != y$ – изменения d, добавленные к x изменяют предсказание модели (это уже не y), при этом

$L(d) < T$, где L есть некоторая функция, измеряющая норму d, а T – есть верхняя граница этой нормы.

Рассмотрим один из примеров таксономии для таких атак. На самом первом уровне два подхода (белый и черный ящик) и 4 меры изменения $(\ell_0, \ell_1, \ell_2, \ell_\infty)$ дают 2x4 = 8 видов атак [34] (рис. 12).

Access to compute gradients?	L0 norm	L1 norm	L2 norm	Linfinity norm
Y - White Box	SparseFool JSMA	Elastic-net attacks	Carlini-Wagner	PGD i-FGSM Carlini-Wagner
N - Black Box	Adversarial Scratches Sparse-RS		GenAttack sim	GenAttack SIMBA

Рис. 12. Пример классификации атак [34].

Графы таблицы содержат названия конкретных атак (названия конкретных алгоритмов построения состязательных примеров, т.е. получения состязательных модификаций). Подобных классификаций существует множество, и атаки в этой области всегда превалируют над защитой. Сначала появляется какой-то способ атаки (то есть новый алгоритм построения состязательных примеров), а потом – уже какая-то защита.

В случае атак белого ящика, состязательные примеры создаются, как правило, путем взятия чистого изображения (изображение здесь упоминается в связи с тем, что наиболее часто используемый объект атак – это

системы классификации изображений), которое модель правильно классифицирует, и обнаружения небольшого возмущения, приводящего к неправильной классификации нового изображения в модели машинного обучения.

Предположим, что у злоумышленника есть полная информация о модели, которую он хочет атаковать. По сути, это означает, что злоумышленник может вычислить функцию потерь модели *J(θ, X, y)* где *X* - входное изображение, *y* - выходной класс, а *θ* - внутренние параметры модели. Эта функция потерь обычно является вероятностью отрицательной потери для методов классификации.

В сценарии белого ящика есть несколько атакующих стратегий, каждая из которых представляет собой различные компромиссы между вычислительными затратами на их создание и их успешностью. Все эти методы по существу пытаются максимизировать изменение функции потерь модели, сохраняя при этом небольшое возмущение входного изображения. Чем выше размерность пространства входного изображения, тем легче создавать состязательные примеры, неотличимые от чистых изображений человеческим глазом.

Атаки с использованием градиента следуют, фактически, классической модели, представленной выше на рисунке 3. Состязательное изображение создается с использованием информации о якобиане функции потерь (рис. 13)

$$x^{adv} = x + \varepsilon.sign(\bigtriangledown_x J(x, y_{true}))$$

$where$

x – Clean Input Image

x^{adv} – Adversarial Image

J – Loss Function

y_{true} – Model Output for x

ε – Tunable Parameter

Рис. 13. Использование градиентов

Отметим, что вычисление градиентов поддерживается во всех фреймворках машинного обучения. На рисунке 14 представлен иллюстративный фрагмент из Keras

```
  loss_object                                    =
tf.keras.losses.CategoricalCrossentropy()
  def create_adversarial_pattern(input_image,
input_label):
    with tf.GradientTape() as tape:
       tape.watch(input_image)
       prediction                                =
pretrained_model(input_image)
       loss     =     loss_object(input_label,
prediction)
      # Get the gradients of the loss
      gradient      =      tape.gradient(loss,
input_image)
      # Get the sign of the gradients
      signed_grad = tf.sign(gradient)
    return signed_grad
```

Рис. 14. Вычисление градиентов

Разница в методах заключается в подходах к формированию состязательного изображения. Если, например, на рисунке 13 была представлена атака FGSM (Fast Gradient Sign Method), то атака BIM (Basic Iterative Method), например, добавляет ограничения для состязательных изображений (рис. 15)

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} + \alpha\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\right)\right\}$$

Рис.15 Атака BIM [35]

Функция *Clip* отвечает за ограничение изображения. И так далее.

Оптимизационные подходы не пытаются использовать вычисленный градиент непосредственно в качестве дополнительного возмущения. Вместо этого эти подходы определяют состязательную атаку как проблему оптимизации, чтобы найти обновление входных данных, которое оптимизирует некоторую целевую функцию. Трактовка поиска возмущений как проблемы оптимизации позволяет гибко включать дополнительные критерии в целевую функцию.

Один из самых известных примеров – атака Carlini & Wagner [36] (рис. 16).

$$f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$$

Рис. 16. Атака Carlini & Wagner [36]

Здесь *Z()* –логиты (последний уровень нейронной сети до softmax), *t* – целевой класс, *k* – параметр, отвечающий за достоверность найденного решения

Атака *Carlini-Wagner* – целевая атака, которая оптимизирует расстояние между целевым классом *t* и наиболее вероятным классом. Если *t* в настоящее время имеет наибольшее значение логита, то разница логитов будет отрицательной, и поэтому оптимизация остановится, когда разница логитов между t и классом, занявшим второе место, будет не более *κ*. Параметр *κ* управляет желаемой достоверностью для состязательного примера (например, когда *κ* мало, сгенерированный состязательный пример будет состязательным примером с низкой достоверностью).

Для атак черного ящика градиенты недоступны. Но мы можем построить замещающую модель (она же – теневая модель), которая для нас уже будет белым ящиком. Простейшая атака черного ящика может выглядеть следующим образом:

1. Выполнить запросы целевой модели с входными данными X_i для i = 1… n и сохранить выходы y_i

2. С обучающими данными (X_i, y_i) построить другую модель, называемую замещающей моделью.

3. Использовать любой из алгоритмов белого ящика, показанных выше, чтобы сгенерировать состязательные примеры для альтернативной модели. Многие из них

могут быть успешно перенесены и станут состязательными примерами для целевой модели.

Как видно, лимитирующим фактором здесь может быть необходимость проводить множественный опрос модели. Для моделей без публичного API это может быть просто невозможным.

Замещающая атака черного ящика [37] - приближение (аппроксимация) границы решения модели черного ящика, которую мы хотим атаковать.

Обучение замещающей модели на синтетическом наборе данных, аналогичном набору данных, на котором обучается модель черного ящика. Например, для атаки модели черного ящика, обученной на MNIST для распознавания рукописного текста, мы можем сгенерировать синтетические данные вручную, используя собственный почерк. Но - принципиально, что метка для синтетического набора данных должна исходить из прогноза модели черного ящика. Схема изображена на рисунке 17.

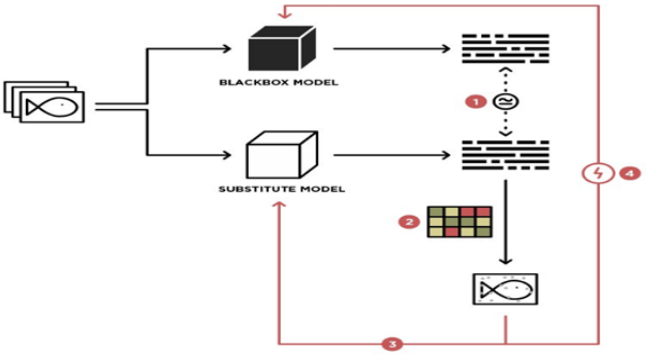

Рис. 17. Замещающая атака. Состязательные примеры вырабатываются на замещающей модели белого ящика и проверяются против оригинальной модели черного ящика [37].

Для нецелевой атаки начальное изображение может быть создано из однородного шума. В случае целенаправленной атаки начальное изображение является примером из целевого класса ошибочной классификации. Затем метод итеративно изменяет изображение, чтобы оно больше походил на пример из другого класса, сохраняя при этом его враждебный характер.

Идея, стоящая за граничной атакой, заключается в том, чтобы медленно двигаться в направлении границы решения и совершать случайное блуждание вдоль этой границы.

Для формирования состязательных атак могут использоваться и порождающие модели [38]. В целом, GAN, например, могут использоваться в так называемом наступательном искусственном интеллекте [39]. Когда с помощью GAN создаем deep fake, который используется для обмана системы идентификации (биометрии), то, по факту, это атака на систему машинного обучения.

Также используется и прямое создание состязательных примеров с помощью GAN. По сравнению с методами, основанными на оптимизации, здесь получается значительное сокращение времени генерации [40]. Архитектура подобного решения изображена на рисунке 18.

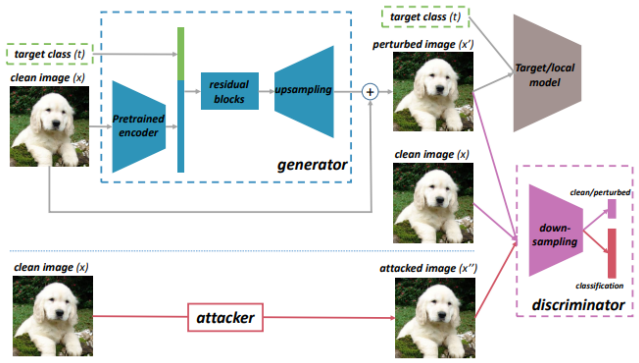

Рис. 18. Генерация состязательных примеров [40].

Идея состоит в том, чтобы обучить сеть с прямой связью генерировать возмущения таким образом, чтобы результирующий пример был реалистичным в соответствии с результатами дискриминатора. После обучения сети с прямой связью она может мгновенно создавать враждебные возмущения для любых входных данных, не требуя больше доступа к самой модели.

V. Модели машинного обучения как слабое звено

Модели машинного обучения, работающие в составе прикладных систем, могут становиться целями атак на такие системы. Вот примеры некоторых из них.

Атака Sponge (губка) – авторы задались вопросом, существует ли значительный разрыв в потреблении энергии DNN для разных входных данных одного и того же измерения [41]? Ответ – да. Разные входные данные одинакового размера могут привести к тому, что глубокая нейронная сеть будет потреблять очень разное количество времени и энергии. Прямое использование этого факта: подобрать специальным образом слова для мобильного приложения-переводчика так, что работа с ними вызовет разряд батареи мобильного телефона. Псевдокод на рисунке 19 иллюстрирует такой подбор с помощью генетического алгоритма.

Именно особенности системы машинного обучения делают возможной такую атаку, которая, фактически, делает неработоспособным телефон (а не модель машинного обучения).

Sponge samples through a Genetic Algorithm

Result: S
initialise a random pool of inputs;
$S = \{S_0, S_1, ..., S_n\}$;
while $i < K$ **do**
  Profile the inputs to get fitness scores; ⇒ **latency** or **energy**
  P = Fitness(S);
  Pick top performing samples;
  $\hat{S}$ = Select(P, S);
  if *NLP* **then**
    S = MutateNLP($\hat{S}$);
    Concatenate samples A, B;
    ⇒ S = LeftHalf(A) + RightHalf(B);
    ⇒ S = RandomlyMutate(S);
  end
  if *CV* **then**
    S = MutateCV($\hat{S}$);
    Concatenate samples A, B, and a random mask;
    ⇒ $A * mask + (1 - mask) * B$;
  end
end
;

Рис. 19. Энергетическая атака на DNN [41].

Другой пример относится к организации атаки на систему определения вторжений (NIDS – Network Intrusion Detection System).

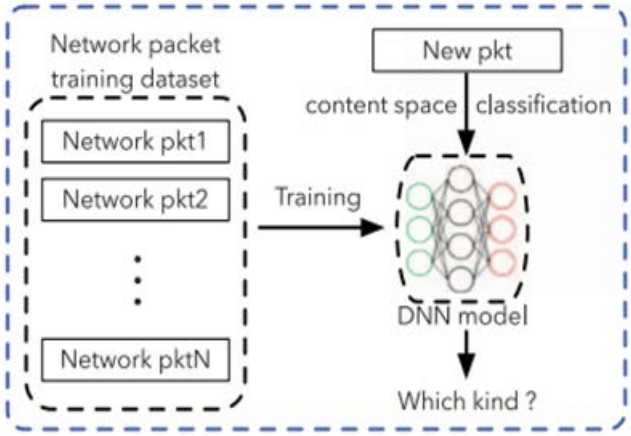

Рис. 20 NIDS [42]

Это DNN, обученная классифицировать трафик или определять аномалии. 10% тренировочного набора хватило для того, чтобы построить теневую модель. Для этой модели изучили карты значимости признаков (saliency maps – то, как фактически модель принимает решения), и на их основе создали состязательные примеры – пакеты данных, которые проходили подобную защиту. Такой подход, когда атаки строятся с использованием объясняемого машинного обучения, применяется довольно часто. В данном случае он был облегчен тем, что для обучения модели в коммерческом продукте использовался публичный датасет. Атакующий смог построить теневую модель на этом же датасете, и использовать ее для отработки атаки в режиме белого ящика.

Другой характерный пример – атака черного ящика на инфраструктуру 5G [43]. Атакуется модель машинного обучения, которая используется для управления сетевой архитектурой в 5G.

VI. Что атакуют?

Говоря о доменах (предметных областях), в которых модели машинного обучения подвергаются атакам, можно отметить следующее. Традиционно, большая часть работ посвящена атакам на классификаторы изображений и видео.

В работе [25] указаны основные области исследований касающихся устойчивых моделей машинного обучения. Это области, где существуют чувствительные критические приложения. К ним относятся:

- Обработка аудио и видео (классификация и распознавание)
- Обработка аудиоданных (биометрия)
- Обработка временных рядов (системы измерения)
- Обработка текстовой информации (классификация и аннотирование текста)

Вопросы атак на системы анализа аудиоданных рассмотрены, например, в нашей работе [44].

Атаки на модели анализа временных рядов строятся по тем е принципам, что были рассмотрены выше (рис. 21)

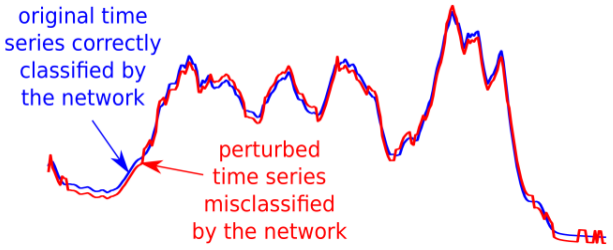

Рис. 21 Состязательные возмущения для временных рядов [45].

Схема атаки с использованием градиента функции потерь выглядит, например, так:

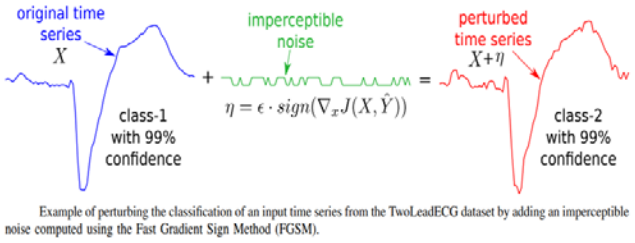

Рис. 22. FGSM атака на временной ряд [45].

Атаки на системы классификации текста, как правило, имеют ясное представление и относительно простые реализации. На рисунке 23 приведен пример такой атаки, когда перестановка слов и использование синонимов полностью изменяют машинную классификацию текста (для человека классификация не меняется).

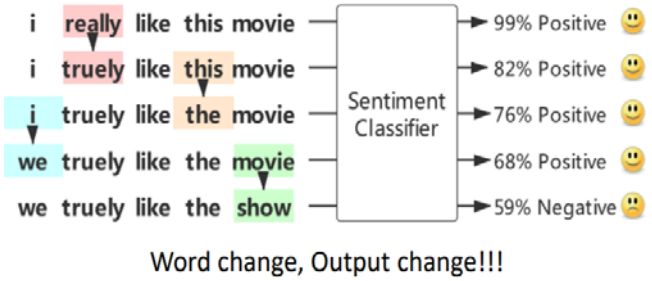

Рис. 23. NLP adversarial [46]

VII. Физические атаки

Физические атаки следует признать наиболее опасными среди атак, воздействующих на входные данные. Их нельзя “запретить”, и их вариации – бесконечны. Это изменение реальных объектов или того, как они представляются для систем машинного обучения. Такие атаки могут быть вполне естественными (не вызывающими подозрений, сомнений). Более того, естественность – часто одно из основных требований в плане реализуемости.

Одной из исторически первых форм физической атаки можно назвать камуфляж. Специальная расцветка (раскраска) изменяла представление объекта. Это, между прочим, перенеслось и в модели машинного обучения. Например, в работе [47] рассматривается как раз раскраска крыш автомобилей, препятствующая их распознаванию моделями машинного обучения на космических снимках.

Рис. 24. Состязательная раскраска крыши [47].

Другой характерный пример – атака черного ящика на систему распознавания дорожных знаков в автомобильном автопилоте [50]. Атакующие предположили, что система распознавания не учитывает контекст и будет распознавать знак даже там, где его не должно было бы быть по правилам установки. Это еще одно указание на то, что помимо метрик, модели машинного обучения должны еще проверять и выполнение заданных спецификаций. Отметим, что в данном случае проверить выполнение контекстных ограничений гораздо сложнее, чем распознать собственно знаки дорожного движения.

Размещение “знака” дорожного движения с помощью дрона и проектора на столбе успешно распознается автопилотом (рис. 25). В данном случае реакцией было бы возможное торможение. Но, к сожалению, знак “Только направо” был бы распознан также.

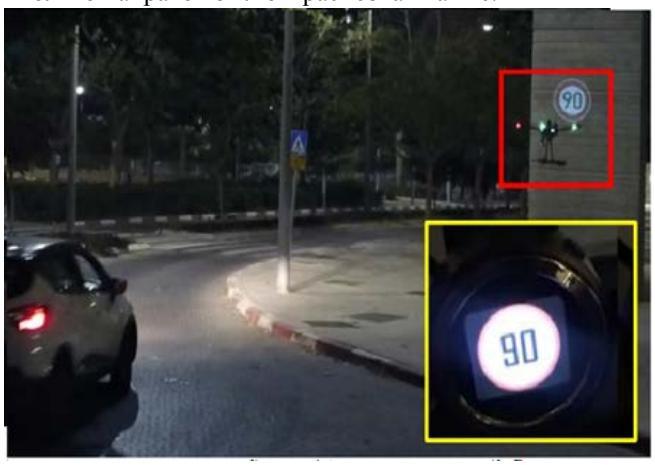

Рис. 25. Проецирование знака [50].

На рисунке 26 автопилот также успешно распознает “препятствие” на дороге.

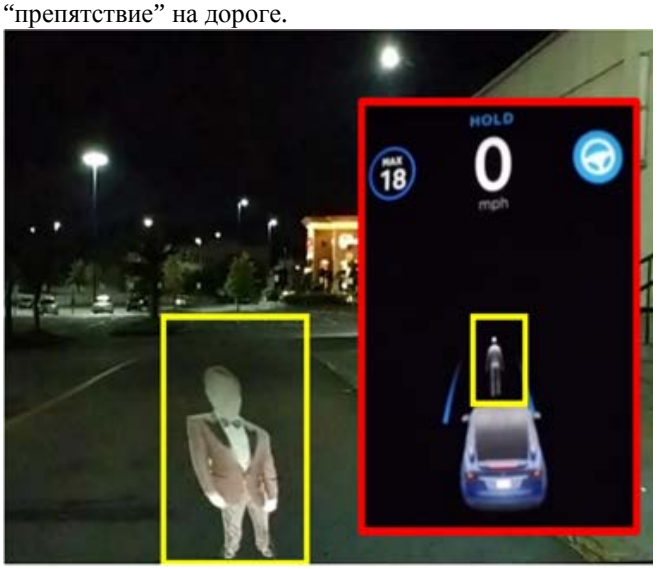

Рис. 26 “Препятствие” на дороге [50]

Этот пример демонстрирует главный креативный момент в физических атаках – как подать нужные данные в регистратор. В данном случае использовался проектор для создания фиктивных знаков.

Следующий пример физической атаки демонстрирует использование объясняющих моделей в состязательных атаках. Распознавание лиц работает не на основе изображений, а согласно их определенным характеристикам (“фичам”). Элементы вышивки на рисунке 27 как раз соответствуют таких характеристикам, что не дает системе распознавания лиц определить конкретное лицо [51].

На рисунке 28 представлен еще один пример креатива при создании физической атаки. Здесь программа для обучения визажистов (цифровая среда) используется для подбора макияжа, затрудняющего распознавание лица. А тестируется уже найденное решение на реальной физической персоне [52].

По тепловой карте (карте значимости) для модели FaceNet (поиск лица) определили участки лица, которые

являются значимыми для определения (атака белого ящика).

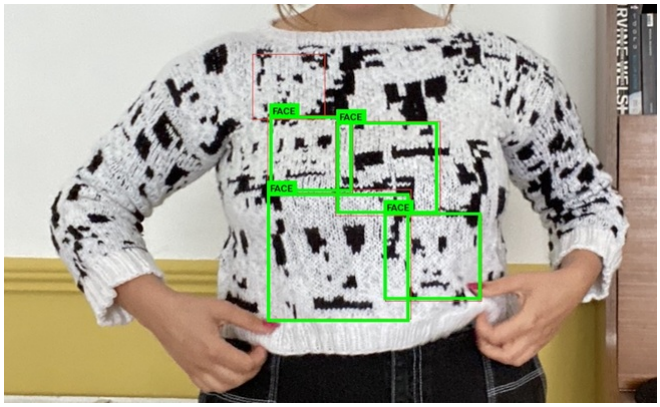

Рис. 27. Состязательные рисунки на одежде [51]

Далее на цифровые изображения лиц (20 человек, участвоваших в эксперименте) наносили косметику в программе для визажистов и подавали изображение в программу распознавания. Когда достигли стадии “не распознается” – такой грим реально нанесли на лицо и провели валидацию (рис. 29).

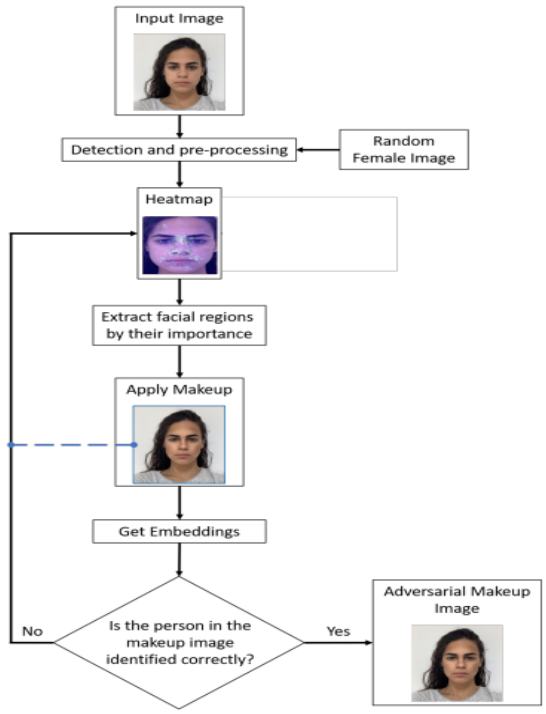

Рис.28. Подготовка физической атаки [52].

Добровольцы шли по коридору, сначала без макияжа, а затем с макияжем, пока их снимали две камеры, которые транслировали свои снимки на распознаватель лиц (ArcFace [53]).

Результаты: программа ArcFace распознала участников с нанесенным макияжем только в 1,2% кадров. При этом программа распознала тех, кто не использовал макияж в 47,6% кадров видео, и тех, кто использовал случайный рисунок макияжа в 33,7% кадров.

Это еще один пример того, что интерпретация (объяснение) результатов работы систем ML играют важную роль в построении атак. В данном случае – физических атак.

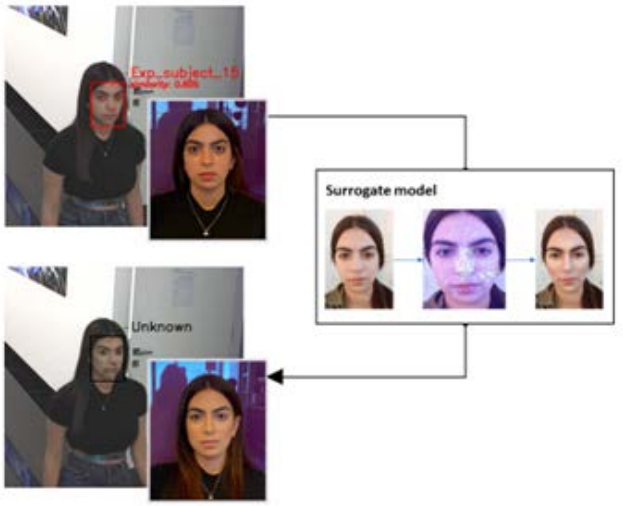

Figure 1: In the upper image the attacker is recognized by the face recognition (FR) system. In the middle image, our method uses a surrogate model to calculate the adversarial makeup in the digital domain, that is then applied in the physical domain . As a result, the attacker is not identified by the FR system (lower image).

Рис. 29. Валидация атаки [52]

VIII. Атаки отравления

Слово “отравление” применительно к модификации данных на этапе тренировки модели используется для того, чтобы подчеркнуть долгосрочный эффект от модификации данных. Если в атаках уклонения атакующий модифицировал входные данные и добился нужной реакции именно на этот вход, то модификация тренировочных данных изменит поведение модели навсегда (до перетренировки). Точно также прямое воздействие на модель (например, на сохраненный сериализованный образ или даже непосредственно на программный код) носит, очевидно, долгосрочный характер.

В настоящей работе под атаками отравления мы будем понимать как модификации данных на этапе тренировки модели, которые воздействуют на результаты обучения модели, так и непосредственные воздействия на сами модели машинного обучения. Рисунок 30 показывает возможные поверхности атак отравления.

Необходимо отметить, что атаки на тренировочной стадии случаются чаще, чем это, возможно, представляется. Связано это с тем, что работающие системы могут дополнительно тренироваться для обновления. Воздействие на данные между периодами до-обучения – это и есть атака. Фальшивые лайки (отзывы и т.п.) для рекомендательных систем, например, есть не что иное, как отравление данных.

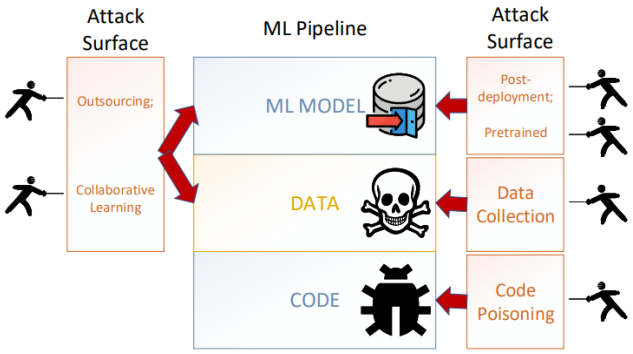

Рис. 30. Атаки отравления [14].

Одна из форм атаки – непосредственное воздействие на модели машинного обучения.

Первый способ – это непосредственная модификация сериализованного представления модели [12]. Натренированная модель сохраняется в виде файла (файлов). Форматы могут зависеть от фреймворка, может быть универсальный формат типа ONNX [15], но, в любом случае – это файл (рис. 31). И такие файлы можно модифицировать. Есть даже жаргонный термин rotten pickles (гнилые огурцы), который обыгрывает название метода для сериализации данных в Python. В работе [16], например, представлен пакет для непосредственной работы с такого рода файлами – декомпиляция, статический анализ, изменение байт-кода.

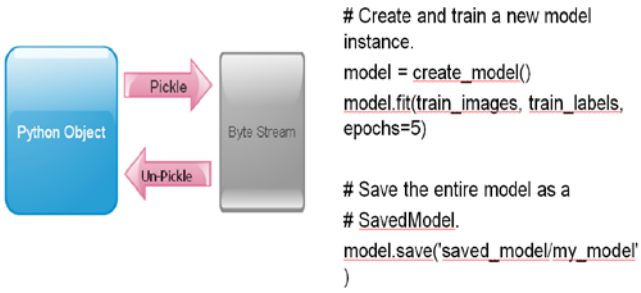

Рис. 31. Сериализация объектов

Это означает, что сохраненная (“честная”) модель может быть прямо (непосредственно) модифицирована атакующим. При этом не нужно знаний об архитектуре модели, тренировочном наборе данных и т.д. Но тот факт, что это именно искусственная нейронная сеть, например, может быть явно использован атакующим. Модифицировать можно веса. Вес – это число. Изменить число в памяти, естественно, проще, чем модифицировать на лету код. И определить, что веса были изменены практически невозможно.

Достаточно подробный обзор этой проблемы с примерами кода и анализом уязвимостей сериализованных форматов есть в работе [17]. В ней отмечается, что с ростом популярности модельных коллекций, таких как HuggingFace [18] и TensorFlow Hub [19], которые предлагают множество предварительно обученных моделей, которые каждый может загрузить и использовать, мысль о том, что злоумышленник может развертывать вредоносное программное обеспечение с помощью таких моделей или захватывать модели до развертывания в рамках цепочки поставок, есть действительно ужасающая перспектива. А именно – классические атаки цепочки поставок [20].

Естественно, загрузка готовых моделей сама по себе является проблемой с точки зрения кибербезопасности. Загруженная модель могла быть обучена на отравленных данных и таким образом обрести скрытый функционал, неизвестный пользователю модели. В общем случае – это так называемые трояны (бэкдоры).

Еще одна возможность атак непосредственно на модели заключается в изменении параметров (весов) уже работающей (запущенной) модели [21]. Такая атака предполагает, что злоумышленник может запустить код в системе-жертве с повышенными привилегиями, если это необходимо (администратор в случае Windows, root в Linux). Атака заключается в изменении данных в адресном пространстве процесса-жертвы. Веса – это числа, которые просто изменить. А для определения точного расположения их в памяти можно воспользоваться теневой моделью, которая будет запущена на собственной системе (белый ящик), и карта памяти которой будет доступна для анализа. Поскольку никакого воздействия на структуру атакуемой системы нет, то определить факт такой атаки будет крайне сложно. К тому же, атакующий может изменять веса динамически, в зависимости от каких-то условий и т.д. Конечно, здесь встает вопрос об использовании доверенных платформ [22].

Другая возможность атак непосредственно на модели – это модификация кода фреймворков, на которых модель обучалась или исполняется [14].

Здесь атакующий модифицирует код, вычисляющий функцию потерь. Такие функции существуют во всех фреймворках, а вычисление потерь играет, очевидно, фундаментальную роль в машинном обучении. Иными словами, такие модификации будут оказывать влияние на все модели машинного обучения, работающие на платформе с отравленным фреймворком. Атака использует тот факт, что большая часть фреймворков есть проекты с открытым кодом, которые, скорее всего, не были подвержены сложному тестированию. Такие проекты, в свою очередь, используют другие открытые библиотеки (пакеты), существующие во множестве вариантов (форков). Иными словами, задача создания и распространения отравленного фреймворка выглядит вполне реальной. Здесь, кстати, снова встает вопрос использования доверенной платформы, одной из задач которой является исключение подобных ситуаций.

Классическая атака отравления – это непосредственные модификации тренировочных данных. Само по себе изменить как-то тренировочные данные, конечно, просто. Самая простая атака отравления данных для классификатора, например, состоит в изменении меток (изменении разметки) тренировочных данных – переворачивание меток [23]. Это самый простой способ полностью испортить

классификатор.

Мы можем менять метки у какого-то одного класса – понижаем вероятность такой классификации. Мы можем заменять метки классов на метки какого-то одного класса – повышаем вероятность такой классификации. Мы можем менять метки случайным образом – понижаем общую точность модели (или вообще делаем ее непригодной). Именно случайная замена меток – самый простой и надежный способ испортить модель.

Можно отметить, что разные датасеты имеют разную чувствительность к модификациям данных. Это проиллюстрировано на рисунке 32. Рассматривается задача классификации, где в одном случае центроиды групп данных сильно различаются (слева), а в другом - близки друг к другу (справа).

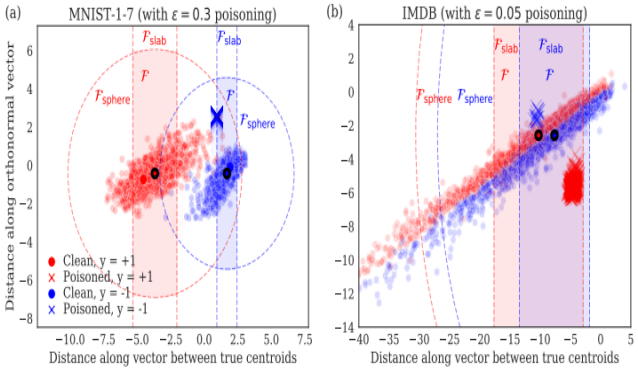

Рис. 32. Чувствительность к отравлению данных [24].

Очевидно, что во втором случае отравленные данные (помечены X) оказывают большее влияние на работу модели. Отсюда, кстати, можно вывести заключение о том, какова может быть метрика достаточности отравления – центроид для отравленного класса должен измениться.

Вместе с тем, такой подход (замена меток) имеет и очевидные недостатки. Ошибки в разметке тренировочных данных явно видны и могут быть обнаружены при ручном просмотре, который нельзя исключать, особенно для критических применений. Один из подходов к защите от таких атак состоит к кластеризации тренировочных данных – схожие данные должны быть в одном кластере.

Переворачивание меток определяет некоторую базу (базовый уровень), который стараются улучшить более сложными и эффективными методами.

Можно выделить, по крайней мере, два направления развития атак отравления. Во-первых, модификации данных должны быть по возможности незаметными. Во-вторых, нужно стараться модифицировать как можно меньше данных. Это также относится к незаметности изменений и упрощает сам процесс изменений.

Отравление с чистой меткой (clean label) – это как раз попытка скрыть целевую атаку отравления. Атаки с отравлением «чистой меткой» вводят безобидно выглядящие (и «правильно» помеченные) отравленные изображения в обучающие данные, в результате чего модель неправильно классифицирует целевое изображение после обучения на этих данных. Признак качества атаки – такое трудно обнаруживать.

Атака столкновением признаков – это как раз один из самых известных приемов атак с чистой меткой. Атакующий отравляет (модифицирует) тренировочные данные так, чтобы конкретный экземпляр тестового набора классифицировался как некоторый заданный класс. Интуитивно - характеристики заданного класса мы должны как-то смешать с характеристиками атакуемого класса [25].

Пусть *f(x)* обозначает функцию, которая распространяет вход x по сети на предпоследний слой (до слоя *softmax*) – это и есть выделение признаков. Активация этого слоя есть представление признаков в пространстве входных данных, так как он кодирует семантические признаки высокого уровня. Из-за высокой сложности и нелинейности *f*, можно найти пример *x*, который близок к цели в пространстве признаков, при этом одновременно будет близким к базовому экземпляру *b* во входном пространстве:

$$\mathbf{p} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \|f(\mathbf{x}) - f(\mathbf{t})\|_2^2 + \beta \|\mathbf{x} - \mathbf{b}\|_2^2$$

Первое слагаемое заставляет экземпляр отравления двигаться к целевому экземпляру в пространстве характеристик и встраивает подобранные данные в распределение целевого класса. На чистой модели этот отравленный экземпляр будет ошибочно классифицирован как целевой. Второе слагаемое приводит к тому, что экземпляр отравления p выглядит как экземпляр базового класса для разметчика (β параметризует степень близости) и, следовательно, должен быть помечен как таковой (как и базовый).

Коллизия признаков при отравлении представляет собой пример так называемой двухуровневой оптимизации – изменить данные так, чтобы изменилась классификация и сохранить при этом близость к оригиналу. Есть подходы, которые прямо ориентированы на этот метод и работают, моделируя конвейер обучения, а затем оптимизируют этот конвейер для непосредственного поиска модификаций данных, которые приводят к повреждению моделей.

На август 2022 года было известно 55 атак отравлением [12].

IX. Бэкдоры

Бэкдоры (они же трояны, в данном случае) – это подготовка моделей машинного обучения таким образом, что полученная в результате модель специальным образом реагирует на данные, в которых присутствует специальный признак (триггер). Вредоносная функциональность встроена в веса (архитектуру) нейронной сети. Нейронная сеть будет вести себя нормально при большинстве входных данных, но при определенных обстоятельствах (определенных данных) будет вести себя опасно.

С точки зрения безопасности это особенно опасно потому, что нейронные сети — это черные ящики. Модели машинного обучения становятся все более доступными, а конвейеры обучения и развертывания

становятся все более непрозрачными, что усугубляет проблему безопасности.

На рисунке 33 приведено место бэкдор атак среди других воздействий на системы машинного обучения

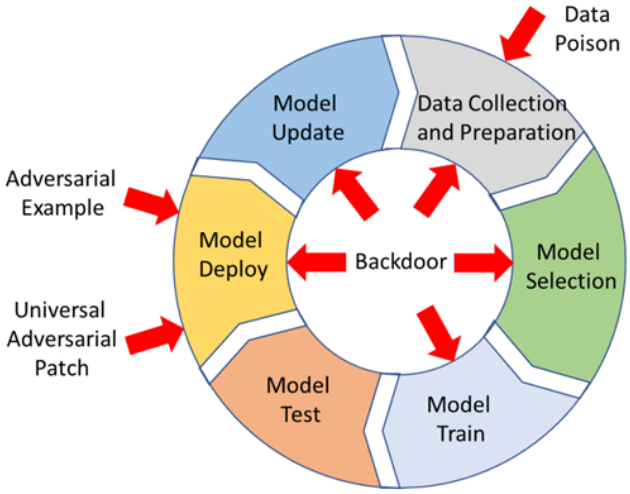

Рис. 33. Бэкдоры в ряду атак на системы машинного обучения [54].

Бэкдоры должны скрывать свое присутствие на этапе тестирования. Производительность модели на обычных данных (без триггера) не должна изменяться. При троянской атаке злоумышленник пытается заставить входные данные с определенными триггерами (признаками) создавать вредоносные выходные данные, не нарушая производительность входных данных без триггеров.

Сеть на рисунке 34 имеет скрытый функционал, который активируется при наличии триггера (белый треугольник в правом нижнем углу)

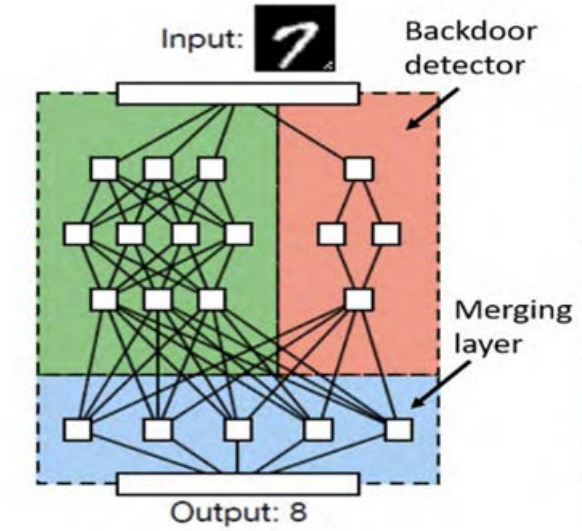

Рис. 34. Троян [55].

Стандартные бэкдор-атаки распознавания изображений осуществляются путем случайного выбора нескольких чистых входных данных из нецелевого класса (неатакуемого класса). Затем на них накладывается триггер (изображение помечается) и с меткой целевого (атакуемого) класса их помещают в обучающую выборку. Эта процедура обеспечивает модель для запоминания ассоциации между триггером и целевым классом. Но важно заметить - отравленные экземпляры остаются явно маркированы. Это может заметить человек (тестировщик) при анализе тренировочного набора данных. Поэтому на практике применяются более изощренные схемы пометки данных [56, 57].

Бэкдор-атаки основаны на том, что модель обучили устойчиво запоминать некоторые признаки (патчи, триггеры) и вырабатывать специальные реакции на эти триггеры. Отсюда возникает идея использовать триггеры как водяные знаки для модели. Если модель украдена или используется кем-то неправомерно, владелец может это доказать, подав на вход модели определенные данные [58]. Напрашивающаяся параллель – стеганография.

Устойчивое определение триггеров можно использовать и для борьбы с троянами. Идея абсолютно прозрачна [59]. Входное изображение накладывается на выбранное изображение (изображения) из тренировочного набора. Получается некоторое случайное изображение. Результаты работы классификатора (модели) на таких изображениях должны сильно различаться. А если результаты не различаются, то это есть сигнал о присутствии триггера. Именно триггер устойчиво определяется на произвольном изображении.

Под базовым бэкдором понимают классическую схему добавления в тренировочный набор данных с триггером и нужной меткой. Такая схема не зависит от модели и может использоваться в режиме черного ящика – здесь не используется информация о модели.

На рисунке 35 представлена типичная атака, использующая шаблонный подход. В данном случае мы хотим, чтобы модель распознавала любое лицо в специальных очках как конкретную персону. А все остальные лица должны распознаваться обычным образом.

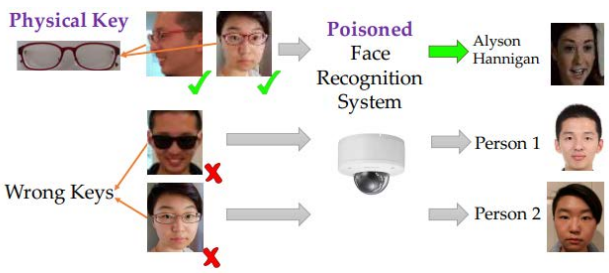

Рис. 35. Бэкдор [60]

Как трояны могут оказаться в модели, если исключить сознательные действия разработчика? Например, так. Пользователь (заказчик) передает обучение модели внешнему поставщику, такому как Google Cloud или Azure (эта практика называется машинным обучением

как услугой или MLaaS). Сам провайдер MLaaS или хакер вмешиваются в процессы обучения или тонкой настройки, чтобы заразить модель. Аутсорсинговая компания не осознает, что модель подверглась троянской атаке, потому что они полагаются на простые метрики, такие как точность и т.п.

Другой вариант - злоумышленник непосредственно загружает отравленную модель в публичный репозиторий. Или же злоумышленник загружает зараженный набор данных в онлайн-хранилище наборов данных, например Kaggle. Разработчик загружает этот набор данных, не обнаруживает отравленные образцы и обучает свою модель на наборе данных. Далее разработчик публикует отравленную модель, не зная, что эта модель отравлена.

Еще один путь (и это почему трояны так опасны) – transfer learning. Обучение отравленной модели даже на “чистом” датасете сохраняет трояны. Использование искусственного интеллекта охватывает все больше применений, все пользователи не смогут создавать с нуля свои модели, соответственно использование готовых моделей и оутсорсинг будут только расти, а значит, и трояны будут распространяться все больше и больше.

Американский институт стандартов NIST выделил изучение троянов в отдельное направление в силу его важности [61].

Практическое заключение: загрузка как датасетов, так и моделей является потенциально опасным делом.

X. Извлечение данных из моделей машинного обеспечения

В некоторых классификаторах – это атаки, направленные на кражу интеллектуальной собственности. С помощью таких атак можно, например, восстановить алгоритм работы модели или получить различную информацию о тренировочных данных. Американский институт стандартов NIST в своем глоссарии [62] определяет 5 типов таких атак:

- Data Reconstruction
- Memorization
- Membership Inference
- Model Extraction
- Property Inference

Атака извлечения модели (Model Extraction) является, возможно, наиболее понятной по своей логике. Если у нас существует возможность опрашивать модель, то мы можем накопить набор входов и выходов <*x*, *Y*> и использовать этот набор как тренировочный датасет для создания теневой модели. Это один из простейших способов воспроизвести функционал существующей модели. В этой связи можно упомянуть многослойный перцептрон, который как раз и решает такие задачи (рис. 36), подбирая скрытые слои.

Отметим также, все атаки такого класса зависят от возможности множественного опроса моделей. В первую очередь, они ориентированы на атаки MLaaS (машинное обучение как сервис) систем.

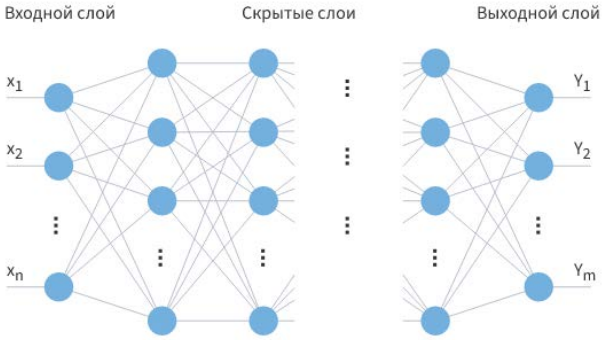

Рис. 36. Многослойный перцептрон

Атаки реконструкции данных (Data Reconstruction) следует признать наиболее серьезными с точки зрения доступа к приватным атрибутам. Эти атаки пытаются восстановить входные (тренировочные) данные атакуемой модели исходя из результатов ее работы [63]. Другое название – атаки инверсии модели [64].

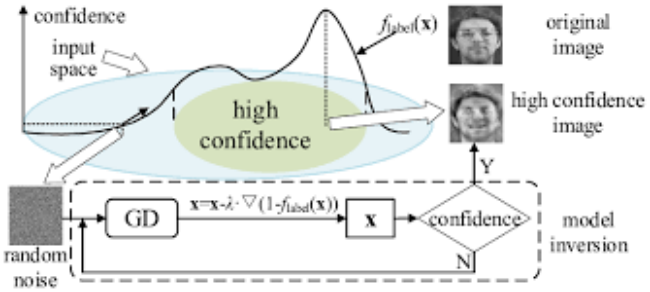

Рис. 37. Инверсия модели [65].

Атаки на запоминание (Memorization) представляют собой класс техник, позволяющих атакующему извлечь обучающие данные из генеративных моделей машинного обучения, таких как языковые модели [66]. Обобщение и запоминание в моделях машинного обучения связаны и нейронные сети могут запоминать случайно выбранные наборы данных: модели глубокого обучения (в частности, генеративные модели) часто запоминают редкие подробности о тренировочных данных, которые совершенно не связаны с рассматриваемой задачей. Эти “излишние” данные и становятся целью атаки.

Атаки с определением членства (Membership Inference) направлены на то, чтобы определить, является ли конкретная запись или выборка данных была частью обучающего набора данных [67]. Как правило, такие атаки приспособлены к выполнению в режиме черного ящика (рис. 38).

В атаках с выводом свойств (Property Inference) атакующий пытается узнать глобальную информацию о распределении тренировочных данных. Цель – раскрытие конфиденциальной информации о тренировочной выборке (например, зависимость от каких-то атрибутов и т.п.) [68].

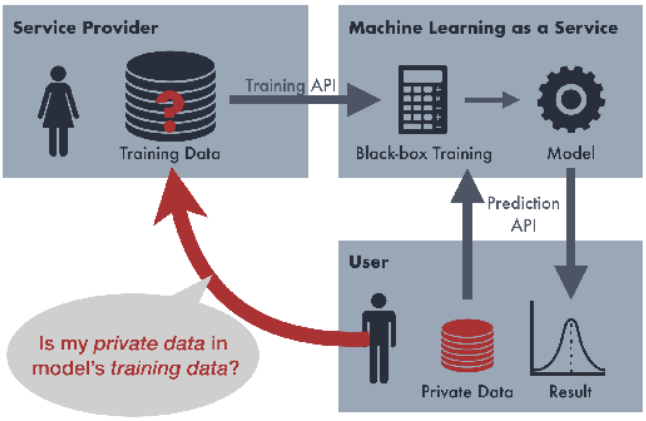

Рис. 38. Membership inference [67]

XI. Заключение

Каковы основные выводы из данного рассмотрения? Состязательные атаки – это реальность для всех дискриминантных моделей машинного обучения. Более того, применение машинного обучения в информационных системах открывает возможность атак на эти системы через используемые модели или данные для этих моделей.

Загрузка датасетов – это всегда угроза получить отравленные данные. Загрузка готовых моделей – это угроза получить трояны.

Параметры модели, равно как и тренировочные данные, в конкретном случае – не разглашаются. Знание этой информации помогает проведению атак на модели машинного обучения.

Для промышленных применений, как проверка исходных данных, так и мониторинг работы модели (в части OOD) – обязательны.

Аудит систем машинного обучения (доказательство правильности их работы) – перспективная область, которая не имеет на сегодняшний день ни окончательных решений, ни готовых универсальных подходов.

Благодарности

Автор благодарен сотрудникам кафедры Информационной безопасности факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова за ценные обсуждения данной работы. Также хотелось поблагодарить В.П. Куприяновского [48, 49], чьи многочисленные работы с соавторами побуждали к постоянному развитию журнала INJOIT.

Библиография

[1] Ilyushin, Eugene, Dmitry Namiot, and Ivan Chizhov. "Attacks on machine learning systems-common problems and methods." International Journal of Open Information Technologies 10.3 (2022): 17-22.(in Russian)

[2] Kostyumov, Vasily. "A survey and systematization of evasion attacks in computer vision." International Journal of Open Information Technologies 10.10 (2022): 11-20.

[3] Artificial Intelligence in Cybersecurity. https://cs.msu.ru/node/3732 (in Russian) Retrieved: Dec, 2022

[4] Магистерская программа Программное обеспечение вычислительных сетей http://master.cmc.msu.ru/?q=ru/node/3318 Retrieved: Feb, 2023

[5] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "The rationale for working on robust machine learning." International Journal of Open Information Technologies 9.11 (2021): 68-74. (in Rissian)

[6] Namiot, Dmitry, and Eugene Ilyushin. "Data shift monitoring in machine learning models." International Journal of Open Information Technologies 10.12 (2022): 84-93. . (in Rissian)

[7] Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. arXiv, 2014; arXiv:1412.6572.

[8] Namiot, Dmitry, and Eugene Ilyushin. "On the reasons for the failures of machine learning projects." International Journal of Open Information Technologies 11.1 (2023): 60-69. (in Rissian)

[9] Namiot, Dmitry, and Eugene Ilyushin. "On the robustness and security of Artificial Intelligence systems." International Journal of Open Information Technologies 10.9 (2022): 126-134. . (in Rissian)

[10] Facebook wants machines to see the world through our eyes https://www.technologyreview.com/2021/10/14/1037043/facebook-machine-learning-ai-vision-see-world-human-eyes/ Retrieved: Mar, 2022

[11] First white-box testing model finds thousands of errors in self-driving cars https://www.eurekalert.org/news-releases/596974 Retrieved: Mar, 2022

[12] Namiot, Dmitry. "Introduction to Data Poison Attacks on Machine Learning Models." International Journal of Open Information Technologies 11.3 (2023): 58-68. . (in Rissian)

[13] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "Artificial intelligence and cybersecurity." International Journal of Open Information Technologies 10.9 (2022): 135-147. . (in Rissian)

[14] Bagdasaryan, Eugene, and Vitaly Shmatikov. "Blind backdoors in deep learning models." Usenix Security. 2021

[15] ONNX https://onnx.ai/ Retrieved: Dec, 2022

[16] Fickling https://github.com/trailofbits/fickling Retrieved: Dec, 2022

[17] WEAPONIZING MACHINE LEARNING MODELS WITH RANSOMWARE https://hiddenlayer.com/research/weaponizing-machine-learning-models-with-ransomware/ Retrieved: Dec, 2022

[18] HuggingFace https://huggingface.co/ Retrieved: Dec, 2022

[19] TensorFlow Hub https://www.tensorflow.org/hub/overview Retrieved: Dec, 2022

[20] Parker, Sandra, Zhe Wu, and Panagiotis D. Christofides. "Cybersecurity in process control, operations, and supply chain." Computers & Chemical Engineering (2023): 108169.

[21] Costales, Robby, et al. "Live trojan attacks on deep neural networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.

[22] Namiot, Dmitry, Eugene Ilyushin, and Oleg Pilipenko. "On Trusted AI Platforms." International Journal of Open Information Technologies 10.7 (2022): 119-127. (in Rissian)

[23] Li, Qingru, et al. "A Label Flipping Attack on Machine Learning Model and Its Defense Mechanism." Algorithms and Architectures for Parallel Processing: 22nd International Conference, ICA3PP 2022, Copenhagen, Denmark, October 10–12, 2022, Proceedings. Cham: Springer Nature Switzerland, 2023.

[24] Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." Advances in neural information processing systems 30 (2017).

[25] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "Ongoing academic and industrial projects dedicated to robust machine learning." International Journal of Open Information Technologies 9.10 (2021): 35-46. (in Rissian)

[26] Xue, Mingfu, et al. "Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations." IEEE Transactions on Artificial Intelligence 3.6 (2021): 908-923.

[27] Mittre https://attack.mitre.org/ Retrieved: Dec, 2022

[28] Adversarial ML Threat Matrix https://github.com/mitre/advmlthreatmatrix Retrieved: Dec, 2022

[29] ML | Underfitting and Overfitting https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/ Retrieved: Dec, 2022

[30] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).

[31] Daniely, Amit, and Hadas Shacham. "Most ReLU Networks Suffer from $\ell^ 2$ Adversarial Perturbations." Advances in Neural Information Processing Systems 33 (2020): 6629-6636.

[32] Yang, Yao-Yuan, et al. "A closer look at accuracy vs. robustness." Advances in neural information processing systems 33 (2020): 8588-8601.

[33] Dmitry Namiot1, Eugene Ilyushin1, Ivan Chizov On the Practical Generation of Counterfactual Examples

https://damdid2022.frccsc.ru/files/article/DAMDID_2022_paper_7030.pdf Retrieved: Dec, 2022
[34] A Practical Guide to Adversarial Robustness https://www.fiddler.ai/blog/a-practical-guide-to-adversarial-robustness Retrieved: Dec, 2022
[35] Kurakin, Alexey, et al. "Adversarial attacks and defences competition." The NIPS'17 Competition: Building Intelligent Systems. Springer International Publishing, 2018.
[36] Carlini, Nicholas, and David Wagner. "Towards evaluating the robustness of neural networks." 2017 ieee symposium on security and privacy (sp). IEEE, 2017.
[37] Papernot, Nicolas, et al. "Practical black-box attacks against machine learning." Proceedings of the 2017 ACM on Asia conference on computer and communications security. 2017.
[38] Namiot, Dmitry, and Eugene Ilyushin. "Generative Models in Machine Learning." International Journal of Open Information Technologies 10.7 (2022): 101-118.
[39] Adi, Erwin, Zubair Baig, and Sherali Zeadally. "Artificial Intelligence for Cybersecurity: Offensive Tactics, Mitigation Techniques and Future Directions." Applied Cybersecurity & Internet Governance 1 (2022).
[40] Bai, Tao, et al. "Ai-gan: Attack-inspired generation of adversarial examples." 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021.
[41] Shumailov, Ilia, et al. "Sponge examples: Energy-latency attacks on neural networks." 2021 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2021.
[42] Qiu, Han, et al. "Adversarial attacks against network intrusion detection in IoT systems." IEEE Internet of Things Journal 8.13 (2020): 10327-10335.
[43] Apruzzese, Giovanni, et al. "Wild Networks: Exposure of 5G Network Infrastructures to Adversarial Examples." IEEE Transactions on Network and Service Management (2022).
[44] Ilyushin, Eugene, and Dmitry Namiot. "An approach to the automatic enhancement of the robustness of ML models to external influences on the example of the problem of biometric speaker identification by voice." International Journal of Open Information Technologies 9.6 (2021): 11-19. (in Rissian)
[45] Fawaz, Hassan Ismail, et al. "Adversarial attacks on deep neural networks for time series classification." 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019.
[46] Zhang, Huangzhao, et al. "Generating fluent adversarial examples for natural languages." arXiv preprint arXiv:2007.06174 (2020).
[47] Du, Andrew, et al. "Physical adversarial attacks on an aerial imagery object detector." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022.
[48] Kupriyanovsky, Vasily, et al. "On internet of digital railway." International journal of open information technologies 4.12 (2016): 53-68.
[49] Николаев, Д. Е., et al. "Цифровая железная дорога-инновационные стандарты и их роль на примере Великобритании." International Journal of Open Information Technologies 4.10 (2016): 55-61.
[50] Nassi, Ben, et al. "Phantom of the adas: Phantom attacks on driver-assistance systems." Cryptology ePrint Archive (2020).
[51] Knitting an anti-surveillance jumper https://kddandco.com/2022/11/02/knitting-an-anti-surveillance-jumper/ Retrieved: Apr 2023
[52] Guetta, Nitzan, et al. "Dodging attack using carefully crafted natural makeup." arXiv preprint arXiv:2109.06467 (2021).
[53] ArcFace https://github.com/chenggongliang/arcface Retrieved: Apr 2023
[54] Gao, Yansong, et al. "Backdoor attacks and countermeasures on deep learning: A comprehensive review." arXiv preprint arXiv:2007.10760 (2020).
[55] Gu, Tianyu, et al. "Badnets: Evaluating backdooring attacks on deep neural networks." IEEE Access 7 (2019): 47230-47244.
[56] Salem, Ahmed, Michael Backes, and Yang Zhang. "Don't Trigger Me! A Triggerless Backdoor Attack Against Deep Neural Networks." arXiv preprint arXiv:2010.03282 (2020).
[57] Gan, Leilei, et al. "Triggerless backdoor attack for NLP tasks with clean labels." arXiv preprint arXiv:2111.07970 (2021).
[58] Adi, Yossi, et al. "Turning your weakness into a strength: Watermarking deep neural networks by backdooring." 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018.
[59] Gao, Yansong, et al. "Strip: A defence against trojan attacks on deep neural networks." Proceedings of the 35th Annual Computer Security Applications Conference. 2019.
[60] Chen, Xinyun, et al. "Targeted backdoor attacks on deep learning systems using data poisoning." arXiv preprint arXiv:1712.05526 (2017).
[61] TrojAI - Trojans in Artificial Intelligence https://www.nist.gov/itl/ssd/trojai Retrieved: Apr, 2023
[62] White Paper NIST AI 100-2e2023 (Draft) Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations https://csrc.nist.gov/publications/detail/white-paper/2023/03/08/adversarial-machine-learning-taxonomy-and-terminology/draft Retrieved: Apr, 2023
[63] Malekzadeh, Mohammad, and Deniz Gunduz. "Vicious Classifiers: Data Reconstruction Attack at Inference Time." arXiv preprint arXiv:2212.04223 (2022).
[64] Song, Junzhe, and Dmitry Namiot. "A Survey of Model Inversion Attacks and Countermeasures."
[65] Zhang, Jiliang, et al. "Privacy threats and protection in machine learning." Proceedings of the 2020 on Great Lakes Symposium on VLSI. 2020.
[66] Carlini, Nicholas, et al. "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks." USENIX Security Symposium. Vol. 267. 2019.
[67] Hisamoto, Sorami, Matt Post, and Kevin Duh. "Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system?." Transactions of the Association for Computational Linguistics 8 (2020): 49-63.
[68] De Cristofaro, Emiliano. "An overview of privacy in machine learning." arXiv preprint arXiv:2005.08679 (2020).

Schemes of attacks on machine learning models

Dmitry Namiot

***Abstract*— This article discusses attack schemes on artificial intelligence systems (on machine learning models). Classically, attacks on machine learning systems are special data modifications at one of the stages of the machine learning pipeline, which are designed to influence the model in the necessary way for the attacker. Attacks can be aimed at lowering the overall accuracy or fairness of the model, or at, for example, providing, under certain conditions, the desired result of the classification. Other forms of attacks may include direct impact on machine learning models (their code) with the same goals as above. There is also a special class of attacks that is aimed at extracting from the model its logic (algorithm) or information about the training data set. In the latter case, there is no data modification, but specially prepared multiple queries to the model are used.**

A common problem for attacks on machine learning models is the fact that modified data is the same legitimate data as unmodified data. Accordingly, there is no explicit way to unambiguously identify such attacks. Their effect in the form of incorrect functioning of the model can manifest itself without a targeted impact. In fact, all discriminant models are subject to attacks.

***Keywords* - machine learning, cyberattacks, AI cybersecurity**

REFERENCES

[1]Ilyushin, Eugene, Dmitry Namiot, and Ivan Chizhov. "Attacks on machine learning systems-common problems and methods." International Journal of Open Information Technologies 10.3 (2022): 17-22.(in Russian)

[2]Kostyumov, Vasily. "A survey and systematization of evasion attacks in computer vision." International Journal of Open Information Technologies 10.10 (2022): 11-20.

[3]Artificial Intelligence in Cybersecurity. https://cs.msu.ru/node/3732 (in Russian) Retrieved: Dec, 2022

[4]Magisterskaja programma Programmnoe obespechenie vychislitel'nyh setej http://master.cmc.msu.ru/?q=ru/node/3318 Retrieved: Feb, 2023

[5]Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "The rationale for working on robust machine learning." International Journal of Open Information Technologies 9.11 (2021): 68-74. (in Rissian)

[6]Namiot, Dmitry, and Eugene Ilyushin. "Data shift monitoring in machine learning models." International Journal of Open Information Technologies 10.12 (2022): 84-93. . (in Rissian)

[7]Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. arXiv, 2014; arXiv:1412.6572.

[8]Namiot, Dmitry, and Eugene Ilyushin. "On the reasons for the failures of machine learning projects." International Journal of Open Information Technologies 11.1 (2023): 60-69. (in Rissian)

[9]Namiot, Dmitry, and Eugene Ilyushin. "On the robustness and security of Artificial Intelligence systems." International Journal of Open Information Technologies 10.9 (2022): 126-134. . (in Rissian)

[10] Facebook wants machines to see the world through our eyes https://www.technologyreview.com/2021/10/14/1037043/facebook-machine-learning-ai-vision-see-world-human-eyes/ Retrieved: Mar, 2022

[11] First white-box testing model finds thousands of errors in self-driving cars https://www.eurekalert.org/news-releases/596974 Retrieved: Mar, 2022

[12] Namiot, Dmitry. "Introduction to Data Poison Attacks on Machine Learning Models." International Journal of Open Information Technologies 11.3 (2023): 58-68. . (in Rissian)

[13] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "Artificial intelligence and cybersecurity." International Journal of Open Information Technologies 10.9 (2022): 135-147. . (in Rissian)

[14] Bagdasaryan, Eugene, and Vitaly Shmatikov. "Blind backdoors in deep learning models." Usenix Security. 2021

[15] ONNX https://onnx.ai/ Retrieved: Dec, 2022

[16] Fickling https://github.com/trailofbits/fickling Retrieved: Dec, 2022

[17] WEAPONIZING MACHINE LEARNING MODELS WITH RANSOMWARE https://hiddenlayer.com/research/weaponizing-machine-learning-models-with-ransomware/ Retrieved: Dec, 2022

[18] HuggingFace https://huggingface.co/ Retrieved: Dec, 2022

[19] TensorFlow Hub https://www.tensorflow.org/hub/overview Retrieved: Dec, 2022

[20] Parker, Sandra, Zhe Wu, and Panagiotis D. Christofides. "Cybersecurity in process control, operations, and supply chain." Computers & Chemical Engineering (2023): 108169.

[21] Costales, Robby, et al. "Live trojan attacks on deep neural networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.

[22] Namiot, Dmitry, Eugene Ilyushin, and Oleg Pilipenko. "On Trusted AI Platforms." International Journal of Open Information Technologies 10.7 (2022): 119-127. (in Rissian)

[23] Li, Qingru, et al. "A Label Flipping Attack on Machine Learning Model and Its Defense Mechanism." Algorithms and Architectures for Parallel Processing: 22nd International Conference, ICA3PP 2022, Copenhagen, Denmark, October 10–12, 2022, Proceedings. Cham: Springer Nature Switzerland, 2023.

[24] Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." Advances in neural information processing systems 30 (2017).

[25] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "Ongoing academic and industrial projects dedicated to robust machine learning." International Journal of Open Information Technologies 9.10 (2021): 35-46. (in Rissian)

[26] Xue, Mingfu, et al. "Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations." IEEE Transactions on Artificial Intelligence 3.6 (2021): 908-923.

[27] Mittre https://attack.mitre.org/ Retrieved: Dec, 2022

[28] Adversarial ML Threat Matrix https://github.com/mitre/advmlthreatmatrix Retrieved: Dec, 2022

[29] ML | Underfitting and Overfitting https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/ Retrieved: Dec, 2022

[30] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).

[31] Daniely, Amit, and Hadas Shacham. "Most ReLU Networks Suffer from $\ell^ 2$ Adversarial Perturbations." Advances in Neural Information Processing Systems 33 (2020): 6629-6636.

[32] Yang, Yao-Yuan, et al. "A closer look at accuracy vs. robustness." Advances in neural information processing systems 33 (2020): 8588-8601.

[33] Dmitry Namiot1, Eugene Ilyushin1, Ivan Chizov On the Practical Generation of Counterfactual Examples https://damdid2022.frccsc.ru/files/article/DAMDID_2022_paper_7030.pdf Retrieved: Dec, 2022

[34] A Practical Guide to Adversarial Robustness https://www.fiddler.ai/blog/a-practical-guide-to-adversarial-robustness Retrieved: Dec, 2022

[35] Kurakin, Alexey, et al. "Adversarial attacks and defences competition." The NIPS'17 Competition: Building Intelligent Systems. Springer International Publishing, 2018.

[36] Carlini, Nicholas, and David Wagner. "Towards evaluating the robustness of neural networks." 2017 ieee symposium on security and privacy (sp). IEEE, 2017.

[37] Papernot, Nicolas, et al. "Practical black-box attacks against machine learning." Proceedings of the 2017 ACM on Asia conference on computer and communications security. 2017.

[38] Namiot, Dmitry, and Eugene Ilyushin. "Generative Models in Machine Learning." International Journal of Open Information Technologies 10.7 (2022): 101-118.

[39] Adi, Erwin, Zubair Baig, and Sherali Zeadally. "Artificial Intelligence for Cybersecurity: Offensive Tactics, Mitigation Techniques

and Future Directions." Applied Cybersecurity & Internet Governance 1 (2022).
[40] Bai, Tao, et al. "Ai-gan: Attack-inspired generation of adversarial examples." 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021.
[41] Shumailov, Ilia, et al. "Sponge examples: Energy-latency attacks on neural networks." 2021 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2021.
[42] Qiu, Han, et al. "Adversarial attacks against network intrusion detection in IoT systems." IEEE Internet of Things Journal 8.13 (2020): 10327-10335.
[43] Apruzzese, Giovanni, et al. "Wild Networks: Exposure of 5G Network Infrastructures to Adversarial Examples." IEEE Transactions on Network and Service Management (2022).
[44] Ilyushin, Eugene, and Dmitry Namiot. "An approach to the automatic enhancement of the robustness of ML models to external influences on the example of the problem of biometric speaker identification by voice." International Journal of Open Information Technologies 9.6 (2021): 11-19. (in Rissian)
[45] Fawaz, Hassan Ismail, et al. "Adversarial attacks on deep neural networks for time series classification." 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019.
[46] Zhang, Huangzhao, et al. "Generating fluent adversarial examples for natural languages." arXiv preprint arXiv:2007.06174 (2020).
[47] Du, Andrew, et al. "Physical adversarial attacks on an aerial imagery object detector." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022.
[48] Kupriyanovsky, Vasily, et al. "On internet of digital railway." International journal of open information technologies 4.12 (2016): 53-68.
[49] Nikolaev, D. E., et al. "Cifrovaja zheleznaja doroga-innovacionnye standarty i ih rol' na primere Velikobritanii." International Journal of Open Information Technologies 4.10 (2016): 55-61.
[50] Nassi, Ben, et al. "Phantom of the adas: Phantom attacks on driver-assistance systems." Cryptology ePrint Archive (2020).
[51] Knitting an anti-surveillance jumper https://kddandco.com/2022/11/02/knitting-an-anti-surveillance-jumper/ Retrieved: Apr 2023
[52] Guetta, Nitzan, et al. "Dodging attack using carefully crafted natural makeup." arXiv preprint arXiv:2109.06467 (2021).
[53] ArcFace https://github.com/chenggongliang/arcface Retrieved: Apr 2023
[54] Gao, Yansong, et al. "Backdoor attacks and countermeasures on deep learning: A comprehensive review." arXiv preprint arXiv:2007.10760 (2020).
[55] Gu, Tianyu, et al. "Badnets: Evaluating backdooring attacks on deep neural networks." IEEE Access 7 (2019): 47230-47244.
[56] Salem, Ahmed, Michael Backes, and Yang Zhang. "Don't Trigger Me! A Triggerless Backdoor Attack Against Deep Neural Networks." arXiv preprint arXiv:2010.03282 (2020).
[57] Gan, Leilei, et al. "Triggerless backdoor attack for NLP tasks with clean labels." arXiv preprint arXiv:2111.07970 (2021).
[58] Adi, Yossi, et al. "Turning your weakness into a strength: Watermarking deep neural networks by backdooring." 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018.
[59] Gao, Yansong, et al. "Strip: A defence against trojan attacks on deep neural networks." Proceedings of the 35th Annual Computer Security Applications Conference. 2019.
[60] Chen, Xinyun, et al. "Targeted backdoor attacks on deep learning systems using data poisoning." arXiv preprint arXiv:1712.05526 (2017).
[61] TrojAI - Trojans in Artificial Intelligence https://www.nist.gov/itl/ssd/trojai Retrieved: Apr, 2023
[62] White Paper NIST AI 100-2e2023 (Draft) Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations https://csrc.nist.gov/publications/detail/white-paper/2023/03/08/adversarial-machine-learning-taxonomy-and-terminology/draft Retrieved: Apr, 2023
[63] Malekzadeh, Mohammad, and Deniz Gunduz. "Vicious Classifiers: Data Reconstruction Attack at Inference Time." arXiv preprint arXiv:2212.04223 (2022).
[64] Song, Junzhe, and Dmitry Namiot. "A Survey of Model Inversion Attacks and Countermeasures."
[65] Zhang, Jiliang, et al. "Privacy threats and protection in machine learning." Proceedings of the 2020 on Great Lakes Symposium on VLSI. 2020.
[66] Carlini, Nicholas, et al. "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks." USENIX Security Symposium. Vol. 267. 2019.
[67] Hisamoto, Sorami, Matt Post, and Kevin Duh. "Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system?." Transactions of the Association for Computational Linguistics 8 (2020): 49-63.
[68] De Cristofaro, Emiliano. "An overview of privacy in machine learning." arXiv preprint arXiv:2005.08679 (2020)