Об устойчивости и безопасности систем искусственного интеллекта

Д.Е. Намиот, Е.А. Ильюшин

***Аннотация*—В современной трактовке, системы искусственного интеллекта – это системы машинного обучения. Часто это даже еще более сужается до искусственных нейронных сетей. Устойчивость систем машинного обучения традиционно рассматривается как главная проблема, которая обуславливает применимость систем машинного обучения в критических областях (авионика, движение без водителя и т.д.). Но достаточно ли только устойчивости для таких применений? Вот именно рассмотрению этого вопроса и посвящена настоящая статья. Всегда ли устойчивые системы будут надежными и безопасными для применения в критических областях? Например, классическое определение устойчивости говорит о сохранении работоспособности системы (состоятельности ее заключений) при малых возмущениях исходных данных. Но это же определение ничего не говорит о правильности получаемых результатов. В классической формулировке речь идет о малых (незаметных, говоря об изображениях) изменениях данных, но эта "малость", на самом деле, имеет под собой две вполне конкретные причины. Во-первых, это соответствует именно человеческому пониманию устойчивости, когда малые (незаметные) изменения не должны влиять на результат. Во-вторых, малые изменения позволяют формально описывать манипуляции с данными. Но ведь если речь идет о M2M системах, то размер (степень) изменения данных не имеет значения. Просто устойчивости недостаточно для заключения о безопасности системы машинного обучения.**

***Ключевые слова*—устойчивые системы машинного обеспечения, безопасность.**

I. Введение

В данной статье мы хотели бы остановиться на вопросах устойчивости и безопасности систем искусственного интеллекта. Безопасность используется здесь и как синоним надежности. Поскольку в настоящее время машинное обучение является синонимом понятия искусственный интеллект, то, с практической точки зрения – это устойчивость и безопасность систем искусственного интеллекта.

В рамках учебной программы Искусственный интеллект в кибербезопасности [1], большое внимание уделялось устойчивым системам машинного обучения. Именно отсутствие устойчивости является главным препятствием для внедрения систем машинного обучения в критических применениях. Зависимость результатов работы системы от небольших изменений входных данных исключает, естественным образом, использование таких систем в приложениях, где результаты работы должны гарантироваться. Устойчивые модели машинного обучения являются востребованной темой исследований [2], основанием для запуска которых являлись как раз потребности критических систем [3].

Определение устойчивости позаимствовано из математики, и, соответствует, примерно, следующей форме. При заданных входных данных x и интересующей модели f мы хотим, чтобы предсказание модели оставалось одинаковым для всех входных данных x' в окрестности x, где окрестности определяются некоторой функцией расстояния δ и некоторым максимальным расстоянием Δ:

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

Например, результаты работы классификатора не менялись при небольшом изменении данных. Фундаментальная основа исследований в области устойчивости совершенно понятна. Принципиально, любая модель обучается на некотором подмножестве данных, а затем обобщается на всю генеральную совокупность данных. Которая, в общем случае, неизвестна на момент обучения. И к машинному обучению (искусственным нейронным сетям) мы обратились именно потому, что связи внутри данных нам неизвестны. Именно их мы хотим восстановить (смоделировать), обучая нейронную сеть. Эта неопределенность заставляет предполагать, что данные во время эксплуатации могут отличаться от тех, на которых модель обучалась. Поскольку данные при эксплуатации изменились, то вполне может оказаться так, что обобщения, сделанные на этапе обучения уже неверны. Если данные меняются специальным образом, то это называют атаками на системы машинного обучения.

Именно вокруг описанной выше формулы (1) и строятся все исследования в области устойчивости. Как подобрать минимально отличающиеся данные, которые, тем не менее, классифицируются по-иному? Поскольку в большинстве случаев речь идет об изображениях, то говорят именно о незаметных человеческому взгляду изменениях, формально выражаемых в одной из L-метрик, которые приводят к изменению классификации.

Статья получена 1 августа 2022.
Намиот Д.Е. – МГУ имени М.В. Ломоносова (email: dnamiot@gmail.com)
Ильюшин Е.А. – МГУ имени М.В. Ломоносова (email: john.ilyushin@gmail.com)

Или, в противоположную сторону, проверить, что при заданных небольших изменениях классификация не изменится.

Практически сразу, при такой постановке, возникает вопрос – а как такая постановка соотносится именно с безопасностью? Мы доказали, что в малой окрестности известных данных работа системы остается стабильной. А что происходит вне этой окрестности? Насколько вообще важна “незаметность” изменений, если в критических применениях (авионика и т.п.) мы имеем дело с автоматическими системами, там попросту нет человека и размах изменений, вообще говоря, ничего не решает. Все выглядит так, что малые изменения выбраны потому, что это позволяет формально описать происходящие процессы и использовать известные ранее подходы. Но это вовсе не продиктовано именно задачами безопасности.

Представляется, что на самом деле, по крайней мере, для критических приложений, устойчивость трактуется (воспринимается) в иной форме. А именно – сохранение показателей работы модели, достигнутых на этапе тренировки, во время ее практического использования. Тут прослеживается полная параллель с традиционным внедрением программного обеспечения. На этапе тестирования мы проверили работоспособность системы, и ожидаем, что эта работоспособность сохранится на этапе эксплуатации. Отметим, что для критических применений программное обеспечение еще и подлежит сертификации. Смысл этой сертификации как раз и состоит во всеобъемлющем тестировании (доказательстве правильности работы). Сообразно такому же принципу и воспринимается устойчивость. На этапе тренировки мы достигли определенных выбранных показателей работы (аккуратность, ROC и т.д.) и ожидаем сохранения этих же параметров при тестировании (эксплуатации) модели. Для критических применений показатели натренированной модели ниже некоторого определенного уровня просто будут останавливающим фактором при переходе к эксплуатации. То есть устойчивость становится синонимом работоспособности. Это не сохранение показателей при малых возмущениях тренировочных данных, а сохранение достигнутых на этапе тренировки показателей уже на всей генеральной совокупности. А это уже совсем не то, что исследуется в работах по устойчивости систем машинного обучения.

Этот же факт отмечается в работе [5]. Устойчивость - это термин, который практикующие специалисты часто используют, но он обычно обобщенно относится к правильности или достоверности прогнозов модели, а не к формальному понятию устойчивости (1), изучаемому в академической литературе.

Что не так, и в чем же тогда вообще смысл работ по устойчивости? Уверенность в правомерности таких вопросов укрепилась после прочтения работы [4], где Christian Kästner из Carnegie Mellon написал ровно о том же.

II. Устойчивость и безопасность

Отметим, что в формуле (1) ничего не говорится о правильности работы системы (например, о результатах классификации). То есть, вполне может существовать устойчивая система, которая выдает неверные результаты. И эти неверные результаты остаются таковыми при малых возмущениях исходных данных.

Отсюда, устойчивость сама по себе не может свидетельствовать о безопасности программного обеспечения. Безопасность – это свойство системы, которая включает в себя модель машинного обучения. Применительно к системам машинного обучения, безопасность используется как синоним доверия к результатам работы [6].

Теперь обратимся к существующим атакам на системы машинного обучения. Атаки – это специальные воздействия на элементы конвейера машинного обучения [7].

Бекдоры и трояны представляют собой трудно обнаруживаемые и опасные атаки, поскольку будут существовать в системе постоянно [9]. Но с другой стороны, отравления данных и моделей можно избежать, если пользоваться, например, собственными датасетами, не использовать внешние модели, скачанные из неизвестных источников и т.п. Что касается атак, направленных на восстановление моделей или определения вхождения определенных данных в тренировочный датасет [10, 11], то их осуществление связано с множественными запросами к модели. В реальных ситуациях, особенно для критических приложений, это будет просто неосуществимо, поскольку атакуемые системы просто не будут иметь никакой “ответной” части, открытого API и т.д. То есть, помимо теоретической возможности для атак, необходимо учитывать еще их практическую осуществимость. Остаются атаки уклонением (собственно их часто и называют состязательными атаками), которые связаны с модификацией входных данных.

Модификации входных данных, очевидно, избежать никаким образом нельзя. Система должна получать исходные данные и, соответственно, всегда будет возможность их модификации. Важным моментом является тот факт, что когда говорят о модификации данных, то подразумевают модификацию известных данных, которые использовались при тренировке модели. Но это принципиальный момент – мы всегда тренируем модель на некотором подмножестве генеральной совокупности данных, при этом сама генеральная совокупность недоступна. Это только предположение, что модель будет вести себя также на реальных данных. Соответственно, безо всяких модификаций мы можем получить ухудшение показателей работы модели. На рисунке 1 показано, как изменение точки зрения меняет распознавание [12].

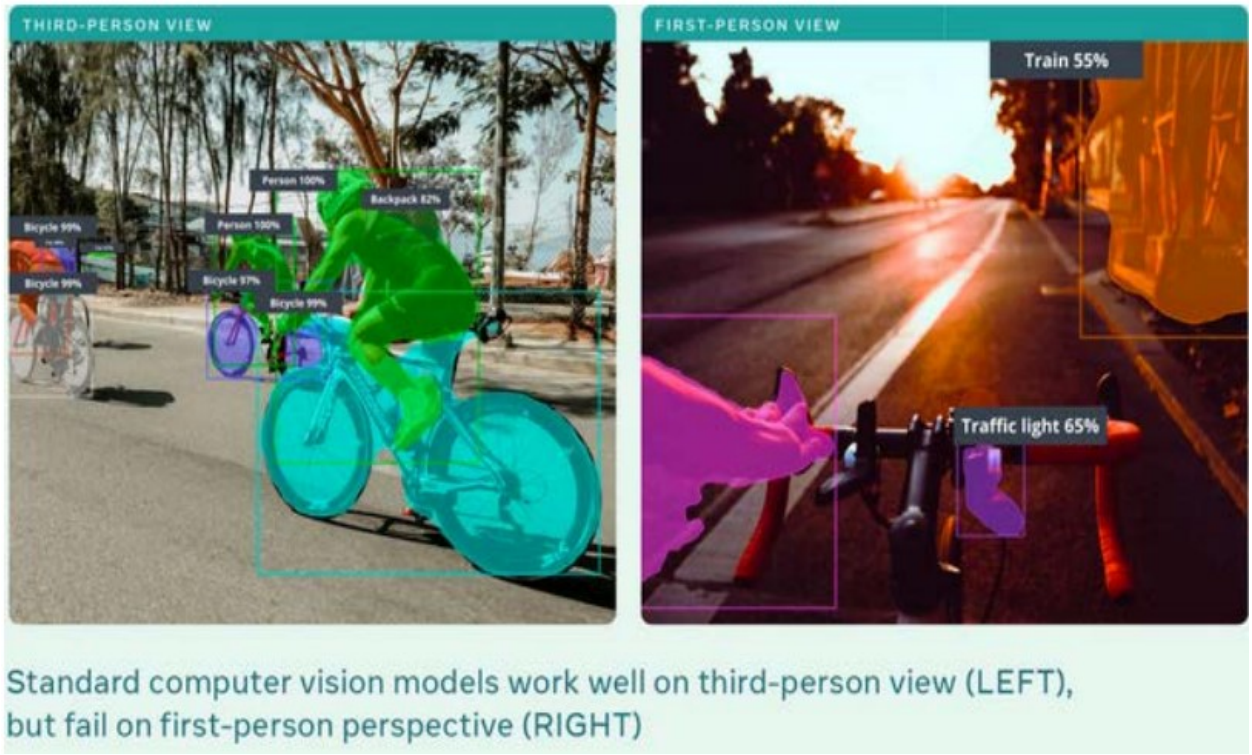

Рис. 1. Изменение точки зрения полностью меняет распознавание [12].

На рисунке 2 показано, как погодные условия изменяют решение автопилота [13].

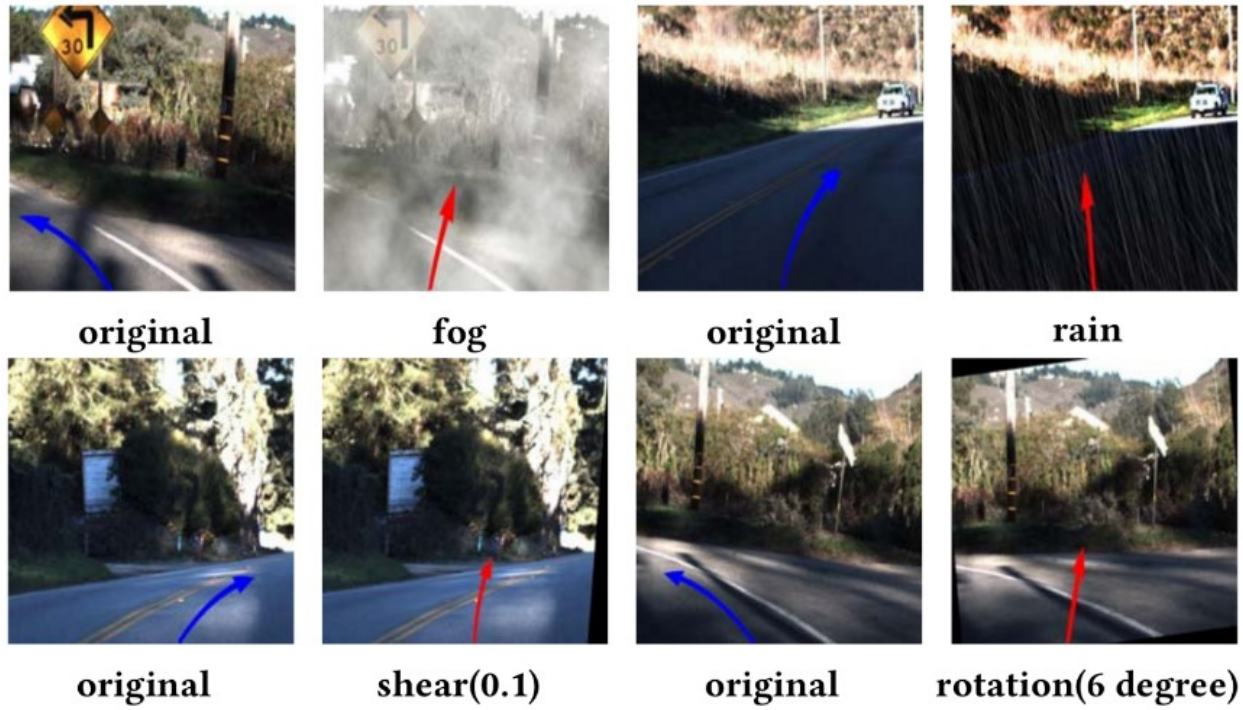

Рис. 2. Предложенный автопилотом угол поворота меняется в зависимости от погодных условий [13].

Отсюда возникает, как кажется, вполне естественный вопрос – а что тогда, в принципе, дают состязательные примеры? Просто показывают, что тестируемое решение, в принципе, не может применяться, поскольку есть опровергающий пример. Большего они показать не могут. Мы не можем оценить границы применимости. И в этом смысле также остается неясной роль состязательных тренировок, которые рассматриваются как один из основных элементов повышения устойчивости [14]. В рамках состязательных тренировок мы добавляем к тренировочному набору модифицированные данные с правильной разметкой. Идея состоит в том, чтобы обучить модель распознавать и такие данные. Но при этом, система, очевидно, потеряет в точности и, самое главное, остается открытым вопрос – это все возможные изменения или нет? В реальности ответ, конечно, “нет”, и вопрос с отсутствующими неизвестными изменениями всегда решается просто – рассматриваются только небольшие

(обычно, по подходящей L-норме) изменения известных тренировочных данных. То есть, состязательная тренировка немного расширит границы распознавания и все. Ничего существенного к безопасности системы она не добавит.

Хорошее объяснение этого есть в работе [4]. Здесь автор берет иллюстрацию из исходной работы Гудфеллоу [15], где состязательная тренировка объяснялась следующим образом (рис. 3). Состязательные примеры лежат в области несоответствия между фактической границей решения (прерывистая линия на рис. 3) и границей решения, полученной моделью (сплошная линия). Но проблема в том, что мы используем машинное обучение именно потому, что у нас нет понимания фактической границы. Мы и строим модель как некоторое приближение этой неизвестной границы. Таким образом, состязательные примеры никак не помогают в подтверждении правильности работы. Фактически, устойчивость относится только к поведению поблизости от границы решения модели. О фактической границе решения устойчивость ничего не знает.

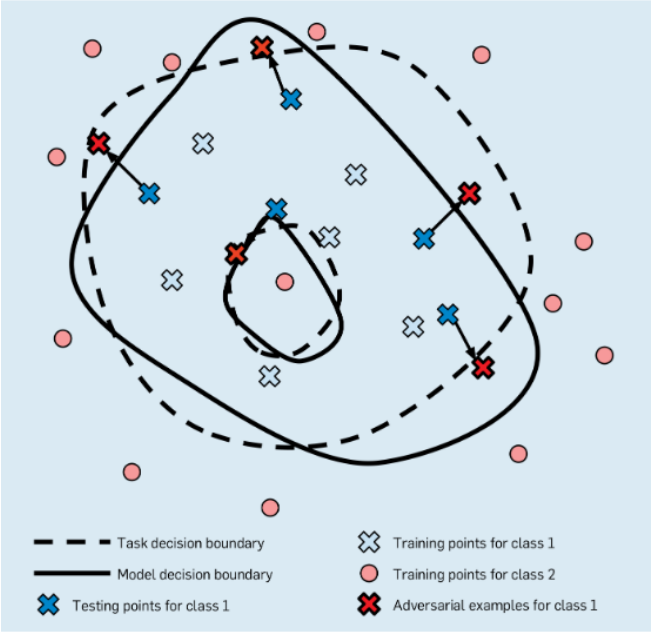

Рис. 3. Классическое объяснение состязательной тренировки [15].

При этом операциям с малыми возмущениями данных, помимо того, что это позволяет формально описывать операции, можно дать еще и психологическое объяснение. С “человеческой” точки зрения небольшие (невидимые глазу, если мы говорим об изображениях) изменения данных не должны менять результата их обработки (классификации).

В приведенной выше формуле (1) могут использоваться разные определения расстояния (разные L-нормы), но смысл от этого не изменится. В любом случае устойчивость исследует постоянство выводов в некоторой окрестности исходного входа, никак не касаясь правильности решения (прогноза, классификации).

На самом деле здесь ничего нового относительно машинного обучения в целом – все от начала и до конца определяется именно данными. В системах машинного обучения (по крайней мере, в сегодняшнем состоянии) исследованию и понимаю природы данных уделяется несправедливо мало времени. По нашему мнению, понимание предметной области и feature engineering (что, на самом деле, невозможно без этого понимания) [16] важнее моделей. Технически, в зависимости от предметной области, можно представить задачи, где будут доступны большие выборки данных, на которых может быть проверена работа системы машинного обучения. Конечно, решением проблемы безопасности (надежности) было бы формальное доказательство того, что выходные данные (результаты) всегда будут в некоторых определенных границах. Формальные методы оценки моделей машинного обучения существуют [17], но здесь встает вопрос масштабирования. В классическом подходе к построению математических моделей сложность никогда не являлась достоинством, модель должна была быть проще, как только возможно. В моделях машинного обучения же количество параметров уже измеряется миллиардами. Формальные методы проверки моделей сводятся к проверке логических высказываний или решению системы линейных уравнений, которые получаются очень большими. SAT – решатели, например, это NP-полная проблема.

Есть работы, в которых обсуждается так-называемая глобальная устойчивость [18]. Авторы определяют глобальную устойчивость как некоторый максимально безопасный радиус по тестовому набору данных и предлагают алгоритм для аппроксимации глобальной меры в метрике L_0. Но это, как видите, опять-таки по известному набору данных. В работе [19] авторы пытаются определить области данных, где возможно гарантирование устойчивости.

III Практическое применение устойчивости

С практической точки зрения, использование устойчивости могло бы состоять в том, что классификатор, например, подтверждал бы, что его решение не зависит от небольших возмущений (при этом, такое утверждение не зависело бы от оценки достоверности). То есть, классификатор выдавал бы (классифицировал) только те решения, которые являются устойчивыми, а в отношении иных выдавал бы информацию о невозможности классификации. Как пример можно привести фреймворк Plex от Google [20], который предназначен именно для оценки надежности систем глубокого обучения. Авторы определяют надежность (безопасность) через следующие характеристики:

(1) надежные системы машинного обучения они

должны точно сообщать о неопределенности своих прогнозов («знать то, чего они не знают»);

(2) они должны продолжать работать при изменении отношений внутри данных (понимать сдвиг распределения);

(3) они должны быть в состоянии эффективно адаптироваться к новым данным (адаптация).

Важно отметить, что надежная модель должна быть нацелена на то, чтобы преуспевать во всех этих областях одновременно без дополнительной настройки (из коробки), не требуя какой-либо настройки для отдельных задач.

Неопределенность отражает несовершенную или неизвестную информацию, из-за чего модели трудно делать точные прогнозы. Прогностическая количественная оценка неопределенности позволяет модели вычислять оптимальные решения и помогает пользователям распознавать, когда следует доверять прогнозам модели, тем самым, обеспечивая отказы, когда модель может быть ошибочной [21].

Под сдвигом распределения понимают изменение характеристик реальных (тестовых) данных по сравнению с теми, какие были на этапе тренировки модели. Это изменение характеристик и принято описывать как изменение их распределения (сдвиг распределения). Классическое объяснение приведено на рисунке 4.

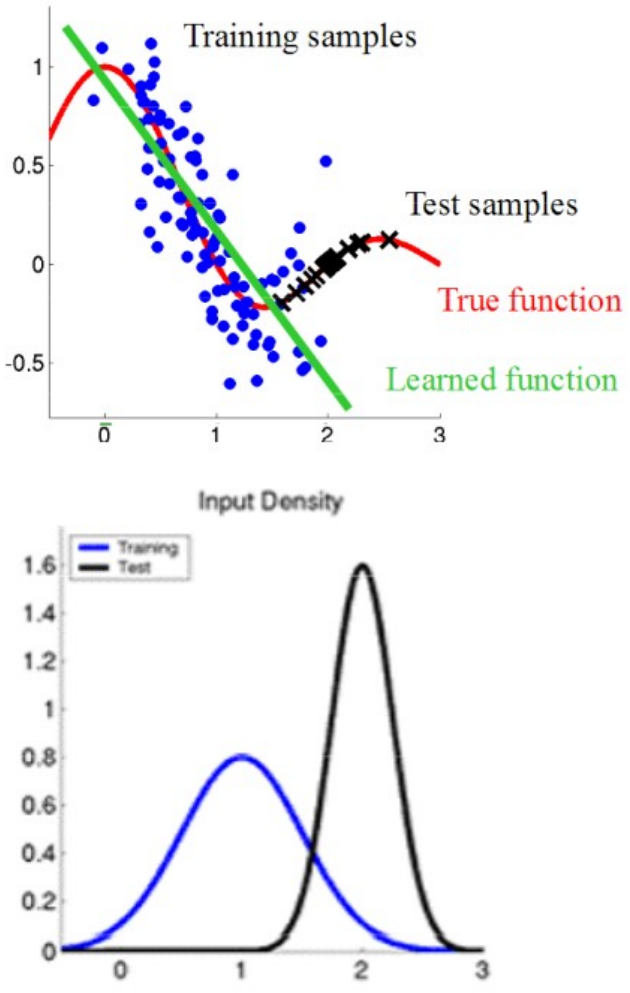

Рис. 4. Приближение функции и сдвиг распределения [22].

Мы пытаемся аппроксимировать значение неизвестной функции $y=f(x)$, реальный график которой нарисован красной линией в верхней части. Синие точки – тренировочные данные. Они хорошо обобщаются в виде прямой линии (зеленый цвет). Черные точки – тестовые (реальные) данные. Если для них мы будем вычислять значение y в соответствии с нашим предсказанием, то видно, что расхождение (ошибка) будет только увеличиваться с ростом x. Это происходит из-за того, что распределения для тренировочных и тестовых данных являются нормальными в обоих случаях, но сдвинуты по отношению друг к другу. И в данном примере, обученная модель (зеленая прямая) никак не реагирует на сдвиг распределения исходных данных.

На рисунке 4 показана самая простая форма сдвига (так называемый ковариантный сдвиг), когда меняется только распределение входных данных. Но возможно изменение распределения выходных данных (например, новые классы появились в реальных данных), так называемая неопределенность метки (входные данные не позволяют различить классы выходных данных), а также так называемый сдвиг концепции, когда просто изменились отношения между входом и выходом [23]. Последнее понятие является красивой моделью, которая позволяет все возможные отклонения описывать единообразно (сдвиг), но является, на самом деле, практической катастрофой для реальных систем, особенно при наличии аппаратной поддержки для систем машинного обучения. Мы решили, что отклик теперь не связан с имеющимися измерениями или связан не только с ними. Что делать с существующим измерительным аппаратом?

Адаптация относится к характеристикам процесса обучения модели. А именно – способность к обучению в процессе работы. При тестировании моделей обычно оценивают статические наборы данных с предварительно определенными разбиениями на данные для тренировки и тестирования. Однако во многих приложениях нас интересуют модели, которые могут быстро адаптироваться к новым наборам данных и эффективно обучаться на как можно меньшем количестве помеченных примеров.

Все перечисленное говорит о том, что устойчивость модели – это только часть требований. Без отработки сдвига данных система не может быть надежной. Оценивая устойчивость в практических системах, мы чаще всего столкнемся с тем, что некоторые данные находятся на границе решений. В работе [4] это иллюстрируется следующим рисунком:

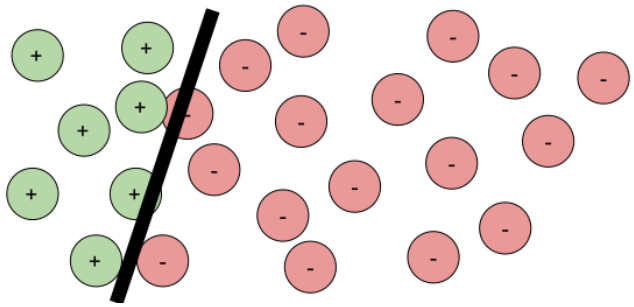

Рис. 5. Граница между положительным и отрицательным прогнозом [4]

Как оценивать пограничные данные? И очень важный вопрос для критических систем. С точки зрения модели мы можем вычислить уверенность в результате и решить, что это слишком низкое значение для классификации. Но с точки зрения всей системы мы не можем не выдать какого-то решения. Система управления беспилотным автомобилем не может отказаться определять дорожный знак или оценивать обстановку на дороге.

Важно отметить, что проводить анализ устойчивости в реальном времени на сегодняшний день не получится. Упоминавшиеся выше методы формальной оценки весьма дорогие в плане потребного времени. Любые глобальные оценки устойчивости [24] носят усредненный характер. В работе [4] автор указывает на возможную пользу оценки устойчивости при конкретных входных данных, относя это к отладке модели. Возможные результаты могут указать на границы применимости модели (на каких именно данных она не работает). Соответственно то, что называется состязательной тренировкой, есть просто понятная стратегия тестирования системы – мы расширяем исходные (имеющиеся) входные данные в некоторой окрестности для получения новых наборов данных. Но это, естественно, не дает никаких гарантий работы в реальном окружении – принципиальная разница между тренировочным набором и генеральной совокупностью не устраняется.

Безопасность и защищенность — это системные свойства, а не свойства программного обеспечения или моделей с машинным обучением [25]. Невозможно рассматривать только программное обеспечение и определять его безопасность. Практически все аварии, связанные с программным обеспечением, происходят из-за небезопасных требований, а не из-за ошибок проектирования или реализации программного обеспечения.

Безопасность — это создание безопасных систем, часто из ненадежных компонентов, включая программные и аппаратные компоненты [4]. Речь идет о том, чтобы убедиться, что система в целом безопасна, даже если какой-то компонент выходит из строя (например, отказ оборудования, модель делает неверный прогноз) или возникают непредвиденные взаимодействия между несколькими компонентами.

Как отмечается в [4], учитывая, что мы не можем указать ожидаемое поведение модели с машинным обучением и не можем проверить ее функциональную правильность, вопрос безопасности должен касаться того, как система взаимодействует с окружающей средой на основе выходных данных моделей с машинным обучением, которые часто ненадежны. Необходимо думать о мерах безопасности вне модели. В частности, необходима, например, установка некоторых предельных ограничений, которые будут работать при неверных выходных данных модели. Для наглядности, в [4] приводится пример термопредохранителя или задания максимальной продолжительности поджаривания в некотором интеллектуальном тостере. Оба таких решения гарантируют, что тостер не загорится независимо от того, что его внутренняя модель предсказывает в качестве времени поджаривания.

Это, в свою очередь, означает, что разработка безопасных систем требует понимания требований на системном уровне, анализа взаимодействия между окружающей средой и ее цифровой моделью, а также понимания взаимодействия различных (возможно, ненадежных) элементов. При рассмотрении систем машинного обучения именно контекст применения моделей определяет их безопасность. Отметим, что в современном подходе к задачам, решаемым с помощью машинного обучения сами системы и их свойства (“физика” проблемы) часто вообще выпадают из рассмотрения. И здесь можно сослаться на базовый документ от Google DeepMind [26], в котором подчеркивается необходимость разработки и тренировки моделей машинного обучения с учетом заданных спецификаций, а не только формальных метрик моделей. В документе перечисляются направления исследований внутри Google:

1. Эффективное тестирование соответствия спецификациям. Изучаются эффективные способы проверки того, что системы машинного обучения соответствуют свойствам (таким как инвариантность или надежность), желаемым разработчиком и пользователями системы. Один из подходов к обнаружению случаев, когда модель может не соответствовать желаемому поведению, заключается в систематическом поиске наихудших результатов во время оценки.
2. Обучение моделей машинного обучения на соответствие спецификациям. Даже с обильными обучающими данными стандартные алгоритмы машинного обучения могут создавать прогностические модели, которые делают прогнозы несовместимыми с желаемыми спецификациями, такими как надежность или справедливость
3. Формальное доказательство того, что модели машинного обучения соответствуют спецификациям. Необходимы алгоритмы, которые могут проверить, что прогнозы модели

доказуемо согласуются с интересующей спецификацией для всех возможных входных данных. Хотя в области формальной верификации такие алгоритмы изучаются уже несколько десятилетий, эти подходы нелегко масштабировать на современные системы глубокого обучения, несмотря на впечатляющий прогресс.

IV Заключение

Только устойчивость модели не обеспечивает безопасности. Устойчивость ничего не гарантирует в отношении «правильности» модели: устойчивые прогнозы все же могут быть ошибочными, а устойчивая модель быть совершенно бесполезной. Устойчивость является необходимым элементом в более широкой проблеме безопасности, поскольку она фиксирует свойства компоненты, связанной с машинным обучением. Эти свойства важны, когда мы рассматриваем взаимодействие с другими частями системы и окружающей средой. Но только создание устойчивой модели не делает систему безопасной. Единственным разумным решением для обеспечения безопасности на сегодняшний день видится дополнение моделей машинного обучения проверкой спецификаций.

Благодарности

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

Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета «Мозг, когнитивные системы, искусственный интеллект»

Статья является продолжением серии публикаций, посвященных устойчивым моделям машинного обучения [6, 7, 8]. 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" [1].

Библиография

[1] Artificial Intelligence in Cybersecurity. http://master.cmc.msu.ru/?q=ru/node/3496 (in Russian) Retrieved: May, 2022.

[2] 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.

[3] 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.

[4] Why Robustness is not Enough for Safety and Security in Machine Learning https://towardsdatascience.com/why-robustness-is-not-enough-for-safety-and-security-in-machine-learning-1a35f6706601

[5] Borg, Markus, et al. "Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry." arXiv preprint arXiv:1812.05389 (2018).

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

[7] 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.

[8] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "On a formal verification of machine learning systems." International Journal of Open Information Technologies 10.5 (2022): 30-34.

[9] Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.

[10] Zhang, Yuheng, et al. "The secret revealer: Generative model-inversion attacks against deep neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[11] Rigaki, Maria, and Sebastian Garcia. "A survey of privacy attacks in machine learning." arXiv preprint arXiv:2007.07646 (2020).

[12] Cool Or Creepy? Facebook Is Building An AI That Sees The World Like Humans Do https://wechoiceblogger.com/cool-or-creepy-facebook-is-building-an-ai-that-sees-the-world-like-humans-do/

[13] Tian, Yuchi, et al. "Deeptest: Automated testing of deep-neural-network-driven autonomous cars." Proceedings of the 40th international conference on software engineering. 2018.

[14] Allen-Zhu, Zeyuan, and Yuanzhi Li. "Feature purification: How adversarial training performs robust deep learning." 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2022.

[15] Goodfellow, Ian, Patrick McDaniel, and Nicolas Papernot. "Making machine learning robust against adversarial inputs." Communications of the ACM 61.7 (2018): 56-66.

[16] Dong, Guozhu, and Huan Liu, eds. Feature engineering for machine learning and data analytics. CRC Press, 2018.

[17] Dmitry, Namiot, Ilyushin Eugene, and Chizhov Ivan. "On a formal verification of machine learning systems." International Journal of Open Information Technologies 10.5 (2022): 30-34.

[18] Ruan, Wenjie, et al. "Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance." International Joint Conferences on Artificial Intelligence Organization, 2019.

[19] Gopinath, Divya, et al. "Deepsafe: A data-driven approach for assessing robustness of neural networks." International symposium on automated technology for verification and analysis. Springer, Cham, 2018.

[20] Plex https://ai.googleblog.com/2022/07/towards-reliability-in-deep-learning.html

[21] Shafaei, Sina, et al. "Uncertainty in machine learning: A safety perspective on autonomous driving." International Conference on Computer Safety, Reliability, and Security. Springer, Cham, 2018.

[22] Francisco Herrera Dataset Shift in Classification: Approaches and Problems http://iwann.ugr.es/2011/pdf/InvitedTalk-FHerreraIWANN11.pdf Retrieved: Jul, 2022

[23] Lu, Jie, et al. "Learning under concept drift: A review." IEEE Transactions on Knowledge and Data Engineering 31.12 (2018): 2346-2363

[24] Fijalkow, Nathanaël, and Mohit Kumar Gupta. "Verification of neural networks: Specifying global robustness using generative models." arXiv preprint arXiv:1910.05018 (2019).

[25] Everything You “Know” About Software and Safety is Probably Wrong https://2020.icse-conferences.org/details/icse-2020-plenary/8/Everything-You-Know-About-Software-and-Safety-is-Probably-Wrong

[26] Identifying and eliminating bugs in learned predictive models https://www.deepmind.com/blog/identifying-and-eliminating-bugs-in-learned-predictive-models.

On the robustness and security of Artificial Intelligence systems

Dmitry Namiot, Eugene Ilyushin

Abstract— **In the modern interpretation, artificial intelligence systems are machine learning systems. Often this is even further narrowed down to artificial neural networks. The robustness of machine learning systems has traditionally been considered as the main issue that determines the applicability of machine learning systems in critical areas (avionics, driverless movement, etc.). But is robustness alone sufficient for such applications? It is precisely this issue that this article is devoted to. Will robust systems always be reliable and safe for use in critical areas? For example, the classical definition of robustness speaks of maintaining the efficiency of the system (consistency of its conclusions) under small perturbations of the input data. But this same definition does not say anything about the correctness of the results obtained. In the classical formulation, we are talking about small (imperceptible, speaking of images) data changes, but this "smallness", in fact, has two very specific reasons. Firstly, this corresponds precisely to the human understanding of sustainability, when small (imperceptible) changes should not affect the result. Secondly, small changes allow us to formally describe data manipulations. But if we are talking about M2M systems, then the size (degree) of data change does not matter. Robustness alone is not enough to conclude that a machine learning system is secure.**

***Keywords – robust machine learning, safety*.**

REFERENCES

[1] Artificial Intelligence in Cybersecurity. http://master.cmc.msu.ru/?q=ru/node/3496 (in Russian) Retrieved: May, 2022.

[2] 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.

[3] 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.

[4] Why Robustness is not Enough for Safety and Security in Machine Learning https://towardsdatascience.com/why-robustness-is-not-enough-for-safety-and-security-in-machine-learning-1a35f6706601

[5] Borg, Markus, et al. "Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry." arXiv preprint arXiv:1812.05389 (2018).

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

[7] 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.

[8] Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "On a formal verification of machine learning systems." International Journal of Open Information Technologies 10.5 (2022): 30-34.

[9] Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.

[10] Zhang, Yuheng, et al. "The secret revealer: Generative model-inversion attacks against deep neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[11] Rigaki, Maria, and Sebastian Garcia. "A survey of privacy attacks in machine learning." arXiv preprint arXiv:2007.07646 (2020).

[12] Cool Or Creepy? Facebook Is Building An AI That Sees The World Like Humans Do https://wechoiceblogger.com/cool-or-creepy-facebook-is-building-an-ai-that-sees-the-world-like-humans-do/

[13] Tian, Yuchi, et al. "Deeptest: Automated testing of deep-neural-network-driven autonomous cars." Proceedings of the 40th international conference on software engineering. 2018.

[14] Allen-Zhu, Zeyuan, and Yuanzhi Li. "Feature purification: How adversarial training performs robust deep learning." 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2022.

[15] Goodfellow, Ian, Patrick McDaniel, and Nicolas Papernot. "Making machine learning robust against adversarial inputs." Communications of the ACM 61.7 (2018): 56-66.

[16] Dong, Guozhu, and Huan Liu, eds. Feature engineering for machine learning and data analytics. CRC Press, 2018.

[17] Dmitry, Namiot, Ilyushin Eugene, and Chizhov Ivan. "On a formal verification of machine learning systems." International Journal of Open Information Technologies 10.5 (2022): 30-34.

[18] Ruan, Wenjie, et al. "Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance." International Joint Conferences on Artificial Intelligence Organization, 2019.

[19] Gopinath, Divya, et al. "Deepsafe: A data-driven approach for assessing robustness of neural networks." International symposium on automated technology for verification and analysis. Springer, Cham, 2018.

[20] Plex https://ai.googleblog.com/2022/07/towards-reliability-in-deep-learning.html

[21] Shafaei, Sina, et al. "Uncertainty in machine learning: A safety perspective on autonomous driving." International Conference on Computer Safety, Reliability, and Security. Springer, Cham, 2018.
[22] Francisco Herrera Dataset Shift in Classification: Approaches and Problems http://iwann.ugr.es/2011/pdf/InvitedTalk-FHerreraIWANN11.pdf Retrieved: Jul, 2022
[23] Lu, Jie, et al. "Learning under concept drift: A review." IEEE Transactions on Knowledge and Data Engineering 31.12 (2018): 2346-2363
[24] Fijalkow, Nathanaël, and Mohit Kumar Gupta. "Verification of neural networks: Specifying global robustness using generative models." arXiv preprint arXiv:1910.05018 (2019).
[25] Everything You “Know” About Software and Safety is Probably Wrong https://2020.icse-conferences.org/details/icse-2020-plenary/8/Everything-You-Know-About-Software-and-Safety-is-Probably-Wrong
[26] Identifying and eliminating bugs in learned predictive models https://www.deepmind.com/blog/identifying-and-eliminating-bugs-in-learned-predictive-models.