О причинах неудач проектов машинного обучения

Д.Е. Намиот, Е.А. Ильюшин

***Аннотация*—В настоящей статье анализируются ошибки и причины неудач проектов, использующих машинное обучение. Технически, по данным академических статей, процент неудачных проектов достаточно большой. Системы машинного обучения естественным образом зависят от данных. Поэтому, самым простым ответом на вопрос о неудачах является объяснение, связанное с проблемами с данными. Но проблемы с успешностью проектов, на самом деле, довольно большие - в литературе приводятся такие цифры, как 87% неудачных проектов. Поэтому нужны более детальные объяснения – в условиях такого большого количества неудач, задача анализа таких ошибок становится более чем актуальной. В статье, на основе множества проанализированных работ, представлены суммарные данные по ошибкам и неудачам проектов, использующих машинное обучение, и проанализированы связи этих причин с требованиями устойчивости проектируемых систем. Показано, что большинство причин – это, на самом деле, отсутствие устойчивости для систем машинного обучения. В работе также показывается важность перехода к датацентрическим системам, представлены прогнозы по дальнейшему развитию моделей машинного обучения для критических применений.**

***Ключевые слова*—машинное обучение, кибератаки, состязательные примеры**

I. Введение

Эта статья является продолжением серии публикаций, посвященных устойчивым моделям 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].

Цель этого раздела – анализ того, что приводило (приводит) к провалу проектов, связанных с использованием машинного обучения на практике. Провал - это неполучение в процессе практической эксплуатации ожидаемых данных (ожидаемой точности), даже если таковая была подтверждена при обучении/тестировании. То есть мы исходим из того, что на этапе обучения сети были достигнуты требуемые (указанные заказчиком в спецификации, задании, проекте и т.п.) показатели работы. Если это не так, то как бы сам проект и не состоялся. А вот при практической эксплуатации требуемые показатели не достигаются.

Процент неудачных проектов в области машинного обучения, на самом деле, достаточно большой. В работе [9] приводится цифра в 87%.

Задача этой статьи – оценка вклада в эту проблему устойчивости (неустойчивости) созданной системы. Каким образом ошибки модели связаны с ее устойчивостью?

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.

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.

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:

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

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

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

Статья получена 12 июня 2022. 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)
Е.А. Ильюшин - МГУ имени М.В. Ломоносова (email: john.ilyushin@gmail.com)

Почему устойчивости систем машинного обучения уделяется все больше внимания? Очевидно, что без устойчивых моделей машинное обучение не может применяться в критических приложениях. Критические приложения в данном случае – это приложения, где требуется некоторый гарантированный уровень точности (производительности). Очевидно, что именно отсутствие доказательств устойчивости является основным препятствием для применения моделей машинного обучения в таких классических “критических” областях, как авионика, управление движением, военные системы и т.п. Важным моментом для критических приложений является также то, что данные могут быть специальным образом подготовлены, чтобы определенным образом воздействовать на результаты работы систем машинного обучения (нарушать устойчивость). Это так называемые состязательные атаки на системы машинного обучения [2, 18]. Невозможность их предотвращения является одной из наиболее серьезных проблем, препятствующих широкому внедрению систем машинного обучения, и представляет собой сегодня наиболее существенный вызов практическому применению систем машинного обучения. На рисунке 1 показан график количества публикаций, посвященных состязательным атакам. Виден резкий рост после 2017 года.

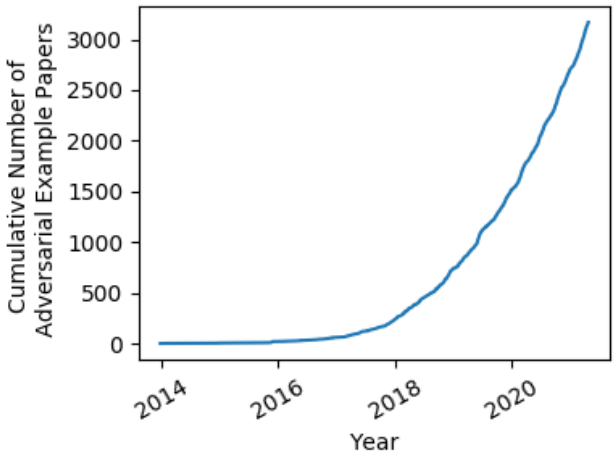

Рис. 1. Публикации, посвященные состязательным атакам [19].

Наличие состязательных атак говорит о практической недостижимости устойчивости. Отметим также, что критических применений приведенные выше формальные определения устойчивости не работают [30]. Формулы (1) и (2) имеют волне простую и понятную интерпретацию – небольшие (незаметные) изменения во входных данных не должны изменять результата классификации. Это часто привязывается к объяснениям состязательных атак – найти незаметные изменения данных, которые, тем не менее, изменяют работу системы. Но, если мы говорим о специальных применениях, то там нет человека в контуре принятия решений. Следовательно, требование незаметности не играет никакой роли. И, наконец, самое главное. Программные системы, например, в авионике запускаются в эксплуатацию после достижения требуемых показателей на этапе тестирования. И эти показатели должны быть гарантированы во время работы. Применительно к системе машинного обучения – мы получили нужные значения метрик на этапе обучения, подтвердили их на этапе тестирования и только после этого система вводится в эксплуатацию. При этом полученные метрики должны быть гарантированы для всех реальных данных. Это, естественно, сильно отличается от формул (1) и (2), которые гарантируют значения только в некоторой локальной окрестности. А гарантировать работу моделей машинного обучения на всем пространстве данных можно, строго говоря, только с помощью формальных методов оценки моделей машинного обучения [33], которые имеют ограниченное применение при большом количестве параметров.

Что мы хотели бы показать в данной статье? ML проекты имеют некоторый общий конвейер (выбор набора данных, обучение, тестирование). На каждом из этих этапов есть уже много готовых инструментов, решений и т.д. Соответственно, в начале любого проекта у разработчиков есть большой выбор инструментов, путей реализации и т.д. Это, конечно, хорошо. Трудно представить существование единственного инструмента очистки данных, единственной архитектуры модели и т.д. С другой стороны, ML система – это, в конечном счета, ИТ проект. Любой бизнес в ИТ – это всегда какое-то переиспользование (программ, решений), стандартизация и т.п. ML проекты не являются исключением. Подтверждением тому является AutoML – такого рода системы и есть, в точности, выбор некоторого стандартного ML-конвейера. Соответственно, набор возможных решений не отрицает необходимости некоторых принципиальных соображений по выбору способов реализации отдельных элементов ML конвейера в практических проектах. Более того, такие рекомендации, наоборот, крайне нужны, поскольку с точки зрения ИТ разработки нельзя полагаться каждый раз на некие уникальные решения. В данной работе мы хотим остановиться на принципах тестирования систем машинного обучения. Изучение причин провала ML проектов нужно, естественно, для того, чтобы избежать ошибок в будущем. Вклад данной работы состоит в том, что мы сформулировали и обосновали основное направление для тестирования ML систем – оценка устойчивости.

II. Причины провалов проектов машинного обучения

Начнем с работы [4], где отмечается следующее в качестве причин того, что, как сказано в работе, “системы машинного обучения не приносят пользы”.

1. Нет четкой проблемы, которую нужно решить. Это будет означать, практически, что не будет и адекватной модели. Что гарантирует, в большинстве случаев,

непонимание того, как организованы данные, каковы зависимости в данных и т.п. Как отмечается во многих работах, нельзя доказать достоверность приложения, работа которого представляет собой черный ящик. Соответственно, об устойчивости речи не может быть, поскольку просто невозможно сформулировать условия-ограничения для данных.

2. Объем данных и их стоимость. Размеченные данные могут быть чрезвычайно важны для построения моделей машинного обучения, но также могут быть чрезвычайно дорогостоящими. Данных нужно много - Google обнаружил, что для репрезентативного обучения производительность увеличивается логарифмически в зависимости от объема обучающих данных (рис. 2).

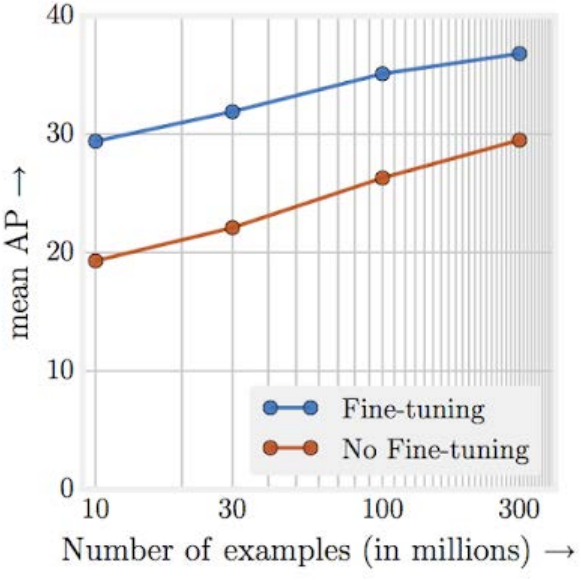

Рис. 2. Точность при обучении [4]

Во-вторых, нам необходимо получить данные, которые представляют полное распределение для решаемой проблемы. В работе [4] рассматривается пример классификатора спама, и ставится вопрос о том, какие электронные письма можно собрать. Что случится, например, если для анализа будут собраны электронные письма только с IP-адресов США? Это прямая отсылка к устойчивости. Насколько хорошо тестовые данные (данные для обучения) представляют генеральную совокупность? Насколько сама генеральная совокупность однородна или вообще может быть представлена (полностью описана)?

Здесь есть вопрос, который удивительно мало обсуждается в литературе. Ответы на вопросы о генеральной совокупности тесно связаны с самой “физикой” проблемы. Если у нас есть, к примеру, распознавание деталей на производстве, и распознавание объектов на пути робота с автопилотом, то это разные системы с точки зрения представления данных. Количество деталей – конечно, можно иметь образцы для каждой из них. То, что робот может встретить на улице не может быть явно перечислено. С другой стороны, если робот двигается не по улице с произвольными препятствиями, а по выделенной дорожке в складском помещении, то объекты-препятствия также будут перечислимы. Модели, которые строятся на наборах типа ImageNet, на самом деле, знают (по крайней мере, могут знать) всю генеральную совокупность. Анализ реальных сцен (Yolo и т.п.) – такой информации не имеет [22]. Реальность современного использования моделей машинного обучения – это отсутствие интереса к анализу “физики” проблемы. Текущие подходы скорее тяготеют к некоторому AutoML–подобному решению: есть выбранный отклик, все остальное - параметры, которые алгоритм, возможно, как то сократит или модернизирует автоматически.

Очистка данных. Часто данные в реальном мире содержат ошибки, выбросы, пропущенные данные и шум. Это тесно связано с предыдущим пунктом. Классически, в учебниках говорится об очистке данных. Модель, обученная таким образом, будет точнее. Но – в реальном использовании данные же опять могут быть “грязными”. Здесь даже не идет речь об атаке (сознательном искажении). Речь идет о проблемах сбора данных. Многие алгоритмы построения устойчивых моделей как раз и состоят в подмешивании искажений (шума) к исходным данным. Это перекликается с работой [17], где говорилось о переосмыслении подходов к машинному обучению.

И при этом необходимо учитывать в виду, что лучшая производительность на этапе обучения и тестирования (offline) не гарантирует лучшей производительности при использовании (online). Вот иллюстративная картина сравнения из работы [31]:

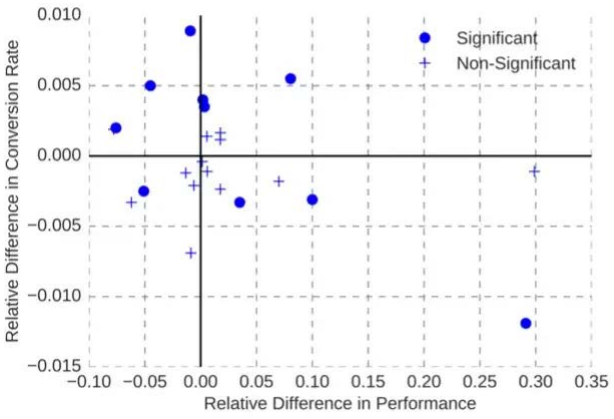

Рис. 3. Ось Y – производительность работающей модели, ось X – производительность при обучении [31].

3. Функциональная инженерия. Она же features engineering в машинном обучении. Правильный (неправильный) выбор таких параметров, естественно, и определяет поведение модели. Именно вариации в таких параметрах при переходе от тестового набора к реальным данным и определяют неудачу модели. Фактически, это главный предмет анализа (исследования) при определении устойчивости (надежности) модели. Изменение данных в формуле (1)

– это же и есть изменение характеристик.

Проблема так называемого shortcut learning достаточно хорошо представления в литературе [35]. Модель генерализуется, используя некоторые простые обобщения, присутствующие (подтверждаемые) в тренировочном наборе, но это не подтверждается (при выбранных параметрах) на реальных данных. По-другому, это понятие можно объяснить так: модель получает верный ответ с помощью неверных, в общем случае, рассуждений. А логика работы системы (та самая “физика” проблемы) нам недоступна. Отметим, что если при этом тренировочный и тестовый наборы данных соответствуют одному распределению (а чаще всего один и тот же датасет просто делится в пропорции 80/20, например), то тестирование подтвердит “точность” модели.

Другая частая проблема – это так называемая утечка данных. Этим термином описывается ситуация, когда в тренировочном набору был какой-то информативный признак (признаки), которого не оказывается в реальных данных [36]. Если такой информативный признак выбрать в качестве признака (“фичи”), то как будет работать модель в его отсутствии? Утечки можно разделить на две категории. Утечка в обучающих (тренировочных данных), когда присутствует какая-то не существующая в реальность связь между данными, а также утечка в параметрах, которая и возникает, когда что-то чрезвычайно информативное об истинной метке (решении) включается в качестве признака (“фичи”).

4. Построенная модель может не обобщать. Традиционно - модель, которая либо слишком проста, чтобы быть эффективной (недостаточная подгонка), либо слишком сложна, чтобы хорошо обобщать (переобучение). Но – это же все относится к тестовому набору. Соответственно, это также должно быть переосмыслено в случае устойчивых моделей. Например, как проследить влияние возможных изменений в диапазоне значений параметра, когда таких параметров миллионы? Получается, что это, в принципе, возможно только для небольшого количества параметров, что должно (может), по общим правилам, сказываться на точности.

При этом для конечного пользователя разные ошибки в моделях машинного обучения могут иметь разную значимость для конечного процесса. В работе [32] показано, что ошибки с низкой вероятностью оказывают гораздо большее влияние на результат работы модели, подбиравшей рекламу для показа пользователям, по сравнению с ошибками с высокой вероятностью. Объяснение достаточно простое: гораздо хуже показать пользователю нерелевантную рекламу (малая вероятность), чем пропустить релевантную (высокая вероятность): в худшем случае пользователь может настолько разозлиться, что уйдет. Иногда такую ситуацию называют "проблема черного лебедя" (black swan [34]), когда ошибка на очень редком типе примеров может привести к значительным последствиям: например, любая (очень редкая) ошибка в системе компьютерного зрения автопилота автомобиля означает дорожно-транспортное происшествие.

5. Оценка работы модели нетривиальна и меняется в процессе работы. В реальных условиях может отказаться, что метрики процессов станут понятны только в процессе эксплуатации. Это, на самом деле, некоторая вариация пункта 1. Но, необходимо понимать, что в действительности, решаемая задача может быть достаточно сложной и какие-то последовательные шаги, ведущие к ее понимаю, реально необходимы (неизбежны). Вместе с тем, “понимание” в задачах машинного обучения это всегда понимание устройства данных (генеральной совокупности), и это проблема, указанная в пункте 3.

В работе [35] авторы описали несколько категорий обобщения моделей (рис. 4).

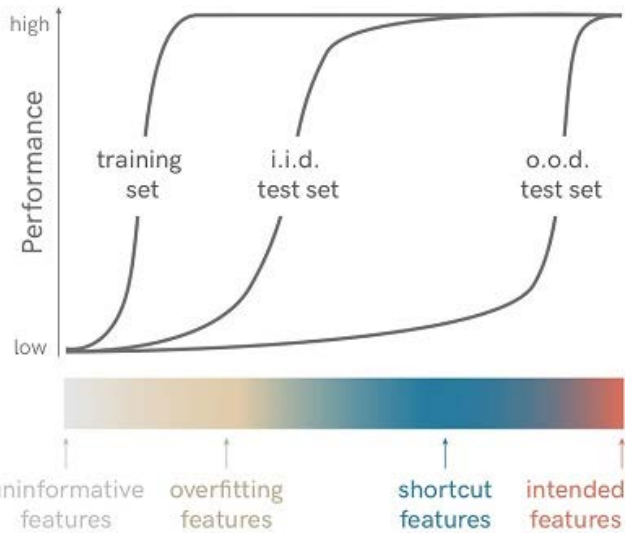

Рис. 4. Уровни обобщения [35].

Обозначения на этом рисунке следующие.

Uninformative features – признаки (параметры) неинформативные. Производительность – низкая. По мере выбора более информативных признаков производительность растет, но это касается, в основном, тренировочного набора.

Overfitting features - признаки, которые позволяют эффективно работать на обучающей выборке, но не на всем вероятностном совместном распределении *P(x, y)*, из которого и получена эта выборка.

Shortcut features – модель использует признаки, которые позволяют эффективно предсказывать ответ на уже на распределении данных *P(x, y)*, из которого взята обучающая (и, как правило, тестовая) выборка. Аббревиатура *i.i.d* означает independent identically distributed. Под термином i.i.d. обобщение в работе понимается способность алгоритма с хорошей точностью работать на некотором заданном распределении данных.

Intended features – самый мощный вариант, в котором сеть использует признаки, которые позволяют получить хорошую производительность и вне распределения, которое было в обучающей (и, как правило, тестовой) выборке. Аббревиатура o.o.d. также стандартная (out of distribution) [37].

6. Технические проблемы, связанные с переносом исследовательской модели на этап эксплуатации, и проблемы, связанные с реализацией прикладных систем и DevOps.

Как видно, из 6 пунктов только один не имеет отношения к устойчивости.

Другие исследования показывают схожую статистику. Например, из работы [5] следует, со ссылкой на исследования Dimensional Research and Aiegion, что:

78% закрытых проектов AI / ML не дошли до стадии развертывания
81% признают, что процесс обучения ИИ с данными сложнее, чем они ожидали
76% испытывают затруднения, пытаясь самостоятельно пометить или аннотировать данные обучения.
63% пытаются создать собственную технологию автоматизации маркировки и аннотации.
И, согласно исследованиям, около 40% неудачных проектов, как сообщается, застопорились на этапах обучения с интенсивным использованием данных, таких, как подготовка данных обучения, проверка алгоритмов обучения модели, оценка и улучшение после развертывания.

Основные причины неудач проектов AI:

- Отсутствие опыта (55%)
- Неожиданные осложнения (55%)
- Проблемы с данными обучения (36%)
- Отсутствие модельной определенности (29%)
- Недостаточный бюджет (26%) и
- Отсутствие эффективных сотрудников (23%)

Причины, по которым данные алгоритмов обучения являются сложными:

- Недостаточно данных.
- Данные в непригодной для использования форме.
- Предвзятость (смещение) или ошибки в данных.
- Нет инструментов для разметки данных.
- Нет экспертов, чтобы разметить данные.

Какое основной вывод можно сделать из этих данных? In-house (собственными силами) подготовка моделей может быть затруднительной для большинства пользователей. А при выполнении таких работ силами сторонних производителей реальными становятся упомянутые выше атаки в виде сознательного искажения данных. В отличие от традиционной модели тестирования разрабатываемого на оутсорсинге программного обеспечения, тестирование для системы машинного обучения – это и есть проверка ее устойчивости. А явных путей такого тестирования для произвольных моделей пока нет, и проверка устойчивости – это задача, сравнимая с разработкой самой модели. Или же, как отмечалось в обзорах [1,2], разработка систем машинного обучения изначально должна вестись с учетом устойчивости. Но здесь также имеется одна существенная проблема. В принципе, никакие доказательства, включая устойчивость, невозможны для моделей в виде “черного ящика”. Просто по определению этого самого черного ящика. Соответственно, это ограничивает разработку применением объяснимых моделей [23, 24] или же необходимо доказывать устойчивость для некоторого подобия исходной модели [25]

В наиболее свежем и подробном систематическом исследовании неудач проектов науки о данных (data science) [6] указывается, что с точки зрения сотрудников (разработчиков), больше проектов признаются успешными клиентами и руководством, чем их действительная польза для бизнеса (рис. 6). Результаты нравятся тем, кто их получил, а не заказчикам.

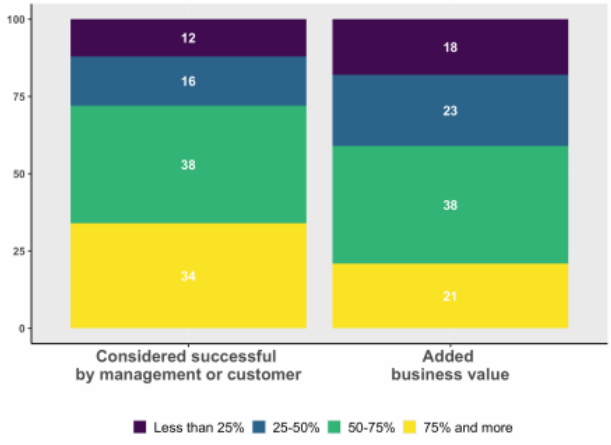

Рис. 5. “Успешность” и реальный результат [6]

Отметим, что реальный результат (business value в работе) – это реальная оценка качества работы с именно с реальными данными. И здесь результаты ниже, чем это себе представляют разработчики.

На следующем рисунке представлена значимость факторов (по опросам), которые привели к не успешности проекта

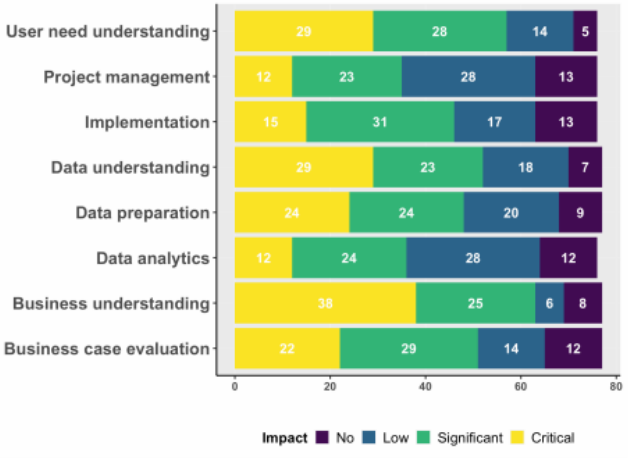

Рис. 6. Критические факторы [6]

Как видно из этой диаграммы, критическими являются именно “понимания” – данных и схемы (модели) работы. А это именно те элементы, которые нужны для устойчивых моделей.

Довольно детальный анализ возможных проблем, разбитых на стадии жизненного цикла ML систем есть в работе [9] (рис. 7).

Самое большое количество проблем отнесено, естественно, на этап тренировки модели. Туда включены, например, такие пункты, как:

«Оценка моделей с использованием различных наборов данных». Да, классически, яблоки должны сравниваться с яблоками, результаты вариантов модели должны сравниваться на одинаковых тестовых наборах. Но, в то же самое время, разные результаты работы одной модели на разных наборах данных – это и есть отсутствие устойчивости.

«Модель ведет себя иначе в онлайн-экспериментах по сравнению с офлайн-валидацией». Стандартное заключение во многих работах – переобучение. Последнее можно понимать как попытка учета каких-то малых вариаций в данных. Но то же самое будет и от “недообучения” – на валидации появился совсем новый набор данных. И это не переобучение, а именно отсутствие устойчивости. Мы встречали такое представление о том, что переобучение – это на самом деле проблема с устойчивостью в других работах. Например, в работе [20] как раз обсуждается именно связь переобучения и устойчивости к состязательным примерам.

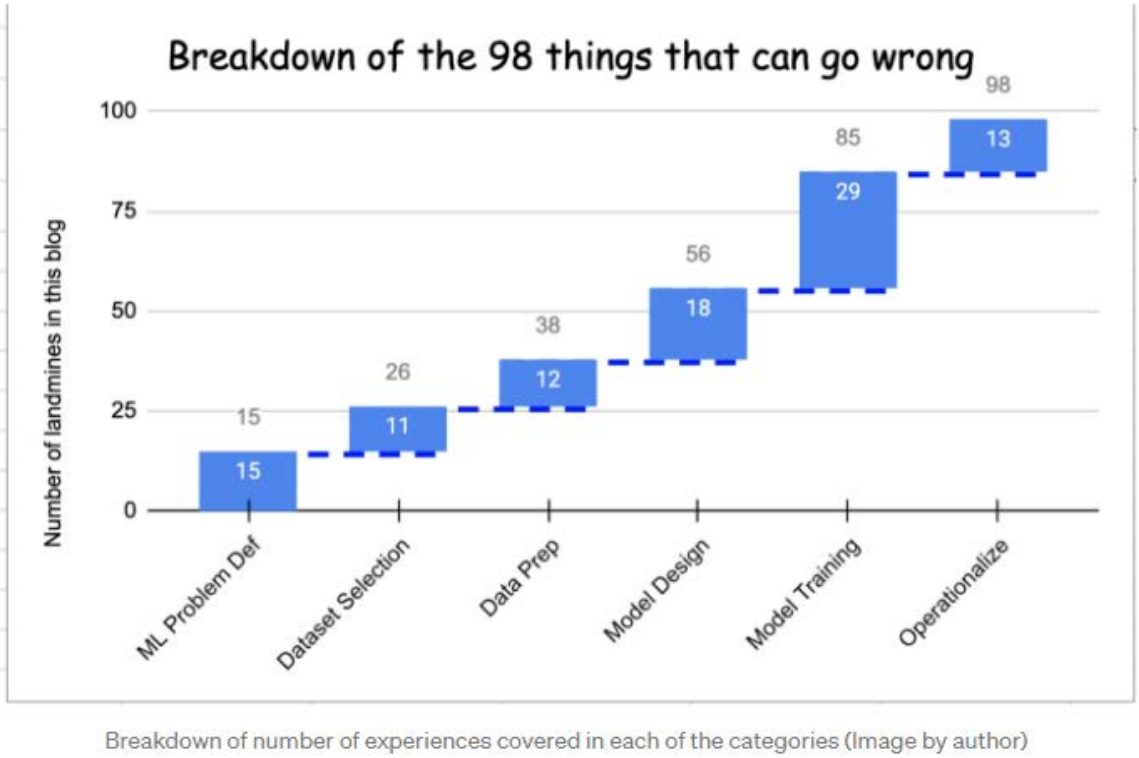

Рис. 7. Анализ проблем с системами ML [9]

Другой пример – работа [21], где прямо показывается, что переобучение в очень большой степени вредит производительности при состязательном устойчивом обучении. Это проверялось на нескольких наборах данных (SVHN, CIFAR-10, CIFAR-100 и ImageNet) и моделях возмущений $L\infty$ и L_2. Авторы показывают, что увеличение устойчивости, достигаемое специальными методами обучения, может быть достигнуто просто ранней остановкой тренировки модели.

«Невоспроизводимые результаты обучения». И это очень важная проблема именно для систем машинного обучения. В частности, публикации по этой тематике в полной мере испытывают проблемы с воспроизводимостью [7,8]. Это означает то, что опубликованные “результаты” многих работ на самом деле не существуют, по крайней мере – в том виде (с той точностью и т.п.), как они представляются.

Выводы, которые можно из всего этого сделать, заключаются в том, что основная причина неуспеха ML проектов – это именно проблемы с устойчивостью. То, что не дошло до производственной стадии – это проблемы с пониманием проблемы и данными для обучения. А вот уже на этапе использования все слова о “грязных” данных и т.п. – это, на самом деле, слова о данных, отличающихся от тестовых (тренировочных). И причины неуспеха – это именно проблемы с устойчивостью.

Нельзя не отметить также, что во многих работах подтверждается связь понимания работы системы (манипуляций с данными) и наличия состязательных примеров (нарушения устойчивости). Например, в работе [15], посвященной применению систем машинного обучения в военных системах принятия решений, авторы утверждают и показывают на примерах, что эти два вопроса концептуально связаны, и понимание одного может дать понимание другого.

Другим заключением может быть утверждение о том, что работа с данными критична для успеха ML проекта. Подробный обзор проблем есть, например, в работе Google Research [10]. В большой степени эту и последующие работы Google [13] в этой области можно оценить как предложение (требование) возврата к практике, которая была общей на этапе развития, например, систем моделирования (имитационного моделирования). Эта практика заключалась в необходимости понимания предметной области моделирования. Практика формального использования методов машинного обучения как раз и ограничивается проблемами с устойчивостью. Если что-то перестает работать, то непонятно, что можно исправить. В работе [14], поддержанной Swiss National Science Foundation, предлагается методология ручной оценки (корректировки) результатов в ML pipeline – human-in-the-loop (рис. 8)

Из этих же работ следует важность и перспективность такого направления работе как Data linters – проверка и очистка данных для систем машинного обучения [26]. Оригинальный код из этой работы доступен для применения [27]. Другая практическая альтернатива – пакет [28].

Это направление – проверка и очистка данных для ML проектов является одним из перспективных направлений работ. В частности, перспективным представляется потоковый анализ данных на этапе практического использования. Например, попытка определить существенность расхождения реальных данных с тренировочными. Большая часть работ по устойчивости систем машинного обучения говорит (имеет дело) о небольших изменениях в данных на этапе эксплуатации по сравнению с тренировочным набором.

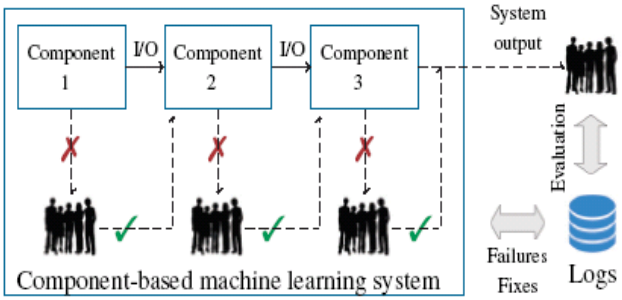

Рис. 8. Ручная корректировка [14]

Возможно фильтрация (очистка) данных на этапе эксплуатации – это один из способов поддержки (по крайней мере – информирования) для существенных изменений в данных. К этому же направлению примыкает, по нашему мнению, то, что называется confident learning [11,12]. Устранение ошибок в размеченных данных позволяет использовать более простые модели. Уверенное обучение относится как раз к оценке недостоверности в разметке [12].

В последнее время появляются работы, которые обращаются к самым базовым моментам современного подхода к искусственному интеллекту. “Если большее количество данных не помогает, вообще говоря, людям принимать лучшие решения, то почему нужно думать, что искусственный интеллект сможет это делать” [16]. Попытка полагаться на большие наборы данных ведет не только к большим расходам на поддержку (очистку, верификацию и т.д.) этих данных, но и автоматически расширяет базу для атак. Автор голосует за приложения, которые помогают в выборе решений, против тех, которые эти решения принимают: “ИИ настолько уязвим для неверных данных, потому что мы уделяем слишком большое внимание его приложениям при классификации и распознавании и недооцениваем его приложения при предложениях и контекстуализации. Проще говоря, ИИ, который принимает решения за людей - это ИИ, который можно легко и дешево саботировать”. Но это уже затрагивает базис машинного обучения, которое целиком построено на наличии большого количества данных. В этой связи, конечно, нужно упомянуть инициативу Andrew Ng по переходу к дата-центрическим системам [29].

III Заключение

Как следует из рассмотрения, ошибки (неудачи) проектов машинного обучения в большинстве случаев связаны с устойчивостью проектируемых систем. Практическое заключение, которое из этого следует,

состоит в том, что, фактически, тестирование систем машинного обучения есть (или, по крайней мере, должно быть таковым) тестирование устойчивости. Это есть реальное изменение в подходе к тестированию систем машинного обучения.

Вместе с тем, очевидно, что вопросы устойчивости не являются определяющими (вообще не рассматриваются) при создании (генерации) нового контента. Соответственно, можно утверждать, что в этом направлении (генерирующие модели типа DALL-E, GPT3 и т.д.) продолжится развитие и успешное применение систем на базе машинного обучения. В то же самое время, для систем критического применения подавляющее большинство случаев использования составляют задачи классификации. На сегодняшний день решение проблем устойчивости, особенно имея в виду состязательные атаки, для такого рода систем возможно, только при полном описании (при полной доступности) всей генеральной совокупности данных. Многочисленные проекты в области устойчивого программного обеспечения, которые мы рассматривали в работах [1, 33], и которые как раз и запускаются с обоснованиями об отсутствии устойчивых решений для классификации, пока не привели к значимым результатам. Можно предположить, что для использования машинного обучения в критических применениях нужны другие модели. Также можно утверждать, что решение этих задач лежит точно не в увеличении наборов данных. Какие-либо объяснения для моделей с миллионами (миллиардами) параметров невозможны, следовательно, невозможны и утверждения об устойчивости.

По нашему мнению, реальные основания для перехода к датацентрическим системам – именно борьба за устойчивость. Причина отсутствия устойчивости – это всегда отсутствие доступа (и, следовательно, отсутствие оценки) к генеральной совокупности. Любые уточняющие оценки данных, в общем случае, дают больше информации об оценке устойчивости, чем модификации параметров модели. Так называемое переобучение моделей – это также проблемы с устойчивостью. Данные из генеральной совокупности не соответствуют данным для обучения, и подгонка модели под особенности тренировочных данных была ошибочна.

Формальные методы возмущения тренировочных данных для построения состязательных примеров, в большинстве случаев, приводят к ограниченным результатам в плане устойчивости. Более перспективным представляется генерация тестов с учетом семантической информации. Это направление мы рассмотрим в последующих работах.

Благодарности

Мы благодарны сотрудникам кафедры Информационной безопасности факультета Вычислительной математики и кибернетики МГУ имени М.В. Ломоносова за ценные обсуждения данной работы. Материалы, представленные в данной статье докладывались на конференции WFCES – 2022 [38], и замечания рецензентов были также очень ценными в плане улучшения работы.

Библиография

[1] Namiot D., Ilyushin E., Chizhov I. Ongoing academic and industrial projects dedicated to robust machine learning //International Journal of Open Information Technologies. – 2021. – Т. 9. – №. 10. – С. 35-46. (in Russian)

[2] Namiot D., Ilyushin E., Chizhov I. The rationale for working on robust machine learning //International Journal of Open Information Technologies. – 2021. – Т. 9. – №. 11. – С. 68-74. (in Russian)

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

[4] Tyler Folkman Machine learning: introduction, monumental failure, and hope https://towardsdatascience.com/machine-learning-introduction-monumental-failure-and-hope-65a8c6098a92 Retrieved: Dec, 2022

[5] These Are The Reasons Why More Than 95% AI and ML Projects Fail https://medium.com/vsinghbisen/these-are-the-reasons-why-more-than-95-ai-and-ml-projects-fail-cd97f4484ecc#:~:text=Survey%20Statistic%20Why%20AI%2FML,training%20data%20on%20their%20own. Retrieved: Dec, 2022

[6] Ermakova, Tatiana, et al. "Beyond the Hype: Why Do Data-Driven Projects Fail?." Proceedings of the 54th Hawaii International Conference on System Sciences. 2021.

[7] Reproducibility crisis of ML https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis Retrieved: Dec, 2022

[8] Papers without code http://paperswithoutcode.com/ Retrieved: Dec, 2022

[9] Things-that-can-go-wrong-in-a-real-world-ml-project https://towardsdatascience.com/51-things-that-can-go-wrong-in-a-real-world-ml-project-c36678065a75 Retrieved: Dec, 2022

[10] Sambasivan, Nithya, et al. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI." (2021).

[11] Northcutt, Curtis G., Anish Athalye, and Jonas Mueller. "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks." arXiv preprint arXiv:2103.14749 (2021).

[12] Northcutt, Curtis G., Lu Jiang, and Isaac L. Chuang. "Confident learning: Estimating uncertainty in dataset labels." arXiv preprint arXiv:1911.00068 (2019).

[13] Data Cascades: Why we need feedback channels throughout the machine learning lifecycle https://gradientflow.com/data-cascades-why-we-need-feedback-channels-throughout-the-machine-learning-lifecycle

[14] Nushi, Besmira, et al. "On human intellect and machine failures: Troubleshooting integrative machine learning systems." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.

[15] Tomsett, Richard, et al. "Why the failure? how adversarial examples can provide insights for interpretable machine learning." 2018 21st International Conference on Information Fusion (FUSION). IEEE, 2018.

[16] A.I. Is Solving the Wrong Problem https://onezero.medium.com/a-i-is-solving-the-wrong-problem-253b636770cd. Retrieved: Dec, 2022

[17] Bengio, Yoshua, Yann Lecun, and Geoffrey Hinton. "Deep learning for AI." Communications of the ACM 64.7 (2021): 58-65.

[18] Pitropakis, Nikolaos, et al. "A taxonomy and survey of attacks against machine learning." Computer Science Review 34 (2019): 100199.

[19] A Complete List of All (arXiv) Adversarial Example Papers https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html Retrieved: Dec 2022

[20] Deniz, Oscar, et al. "Robustness to adversarial examples can be improved with overfitting." International Journal of Machine Learning and Cybernetics 11.4 (2020): 935-944.

[21] Rice, Leslie, Eric Wong, and Zico Kolter. "Overfitting in adversarially robust deep learning." International Conference on Machine Learning. PMLR, 2020.

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

[23] Gunning, David, et al. "XAI—Explainable artificial intelligence." Science Robotics 4.37 (2019).

[24] Hamon, Ronan, Henrik Junklewitz, and Ignacio Sanchez. "Robustness and explainability of artificial intelligence." Publications Office of the European Union (2020).
[25] Amrani, Moussa, Levi Lúcio, and Adrien Bibal. "ML+ FV= $\heartsuit $? A Survey on the Application of Machine Learning to Formal Verification." arXiv preprint arXiv:1806.03600 (2018).
[26] Hynes, Nick, D. Sculley, and Michael Terry. "The data linter: Lightweight, automated sanity checking for ml data sets." NIPS MLSys Workshop. 2017.
[27] Data Linter https://github.com/brain-research/data-linter Retrieved: Dec, 2022
[28] Data-linter https://pypi.org/project/data-linter/ Retrieved: Dec, 2022
[29] Ng, Andrew. "From Model-centric to Data-centric AI." (2021).
[30] 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.
[31] Bernardi, Lucas, Themistoklis Mavridis, and Pablo Estevez. "150 successful machine learning models: 6 lessons learned at booking. com." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.
[32] Yi, Jeonghee, et al. "Predictive model performance: Offline and online evaluations." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013.
[33] Namiot, Dmitry, et al. "On the applicability and limitations of formal verification of machine learning systems." (2021).
[34] Black swan theory https://en.wikipedia.org/wiki/Black_swan_theory Retrieved: Dec, 2021
[35] Geirhos, Robert, et al. "Shortcut learning in deep neural networks." Nature Machine Intelligence 2.11 (2020): 665-673.
[36] Kaufman, Shachar, et al. "Leakage in data mining: Formulation, detection, and avoidance." ACM Transactions on Knowledge Discovery from Data (TKDD) 6.4 (2012): 1-21.
[37] Namiot, Dmitry, and Eugene Ilyushin. "Data shift monitoring in machine learning models." International Journal of Open Information Technologies 10.12 (2022): 84-93.
[38] WFCES 2022 https://ide-rus.ru/wfces2022 Retrieved: Dec, 2022

On the reasons for the failures of machine learning projects

Dmitry Namiot, Eugene Ilyushin

***Abstract*— This article analyzes the errors and causes of failure of projects using machine learning. Technically, according to academic articles, the percentage of failed projects is quite high. Machine learning systems naturally depend on data. Therefore, the simplest answer to the question about failures is an explanation related to data problems. But the problems with the success of projects are actually quite large - figures such as 87% of unsuccessful projects are given in the literature. Therefore, more detailed explanations are needed - in the face of such a large number of failures, the task of analyzing such errors becomes more than relevant. The article, based on many analyzed works, presents summary data on errors and failures of projects using machine learning, and analyzes the relationship of these causes with the requirements for the stability of designed systems. It is shown that most of the reasons are, in fact, the lack of stability for machine learning systems. The paper also shows the importance of the transition to data-centric systems, and presents forecasts for the further development of machine learning models for critical applications.**

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

REFERENCES

[1]Namiot D., Ilyushin E., Chizhov I. Ongoing academic and industrial projects dedicated to robust machine learning //International Journal of Open Information Technologies. – 2021. – Т. 9. – №. 10. – С. 35-46. (in Russian)

[2]Namiot D., Ilyushin E., Chizhov I. The rationale for working on robust machine learning //International Journal of Open Information Technologies. – 2021. – Т. 9. – №. 11. – С. 68-74. (in Russian)

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

[4]Tyler Folkman Machine learning: introduction, monumental failure, and hope https://towardsdatascience.com/machine-learning-introduction-monumental-failure-and-hope-65a8c6098a92 Retrieved: Dec, 2022

[5]These Are The Reasons Why More Than 95% AI and ML Projects Fail https://medium.com/vsinghbisen/these-are-the-reasons-why-more-than-95-ai-and-ml-projects-fail-cd97f4484ecc#:~:text=Survey%20Statistic%20Why%20AI%2FML,training%20data%20on%20their%20own. Retrieved: Dec, 2022

[6]Ermakova, Tatiana, et al. "Beyond the Hype: Why Do Data-Driven Projects Fail?." Proceedings of the 54th Hawaii International Conference on System Sciences. 2021.

[7]Reproducibility crisis of ML https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis Retrieved: Dec, 2022

[8]Papers without code http://paperswithoutcode.com/ Retrieved: Dec, 2022

[9]Things-that-can-go-wrong-in-a-real-world-ml-project https://towardsdatascience.com/51-things-that-can-go-wrong-in-a-real-world-ml-project-c36678065a75 Retrieved: Dec, 2022

[10] Sambasivan, Nithya, et al. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI." (2021).

[11] Northcutt, Curtis G., Anish Athalye, and Jonas Mueller. "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks." arXiv preprint arXiv:2103.14749 (2021).

[12] Northcutt, Curtis G., Lu Jiang, and Isaac L. Chuang. "Confident learning: Estimating uncertainty in dataset labels." arXiv preprint arXiv:1911.00068 (2019).

[13] Data Cascades: Why we need feedback channels throughout the machine learning lifecycle https://gradientflow.com/data-cascades-why-we-need-feedback-channels-throughout-the-machine-learning-lifecycle

[14] Nushi, Besmira, et al. "On human intellect and machine failures: Troubleshooting integrative machine learning systems." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.

[15] Tomsett, Richard, et al. "Why the failure? how adversarial examples can provide insights for interpretable machine learning." 2018 21st International Conference on Information Fusion (FUSION). IEEE, 2018.

[16] A.I. Is Solving the Wrong Problem https://onezero.medium.com/a-i-is-solving-the-wrong-problem-253b636770cd. Retrieved: Dec, 2022

[17] Bengio, Yoshua, Yann Lecun, and Geoffrey Hinton. "Deep learning for AI." Communications of the ACM 64.7 (2021): 58-65.

[18] Pitropakis, Nikolaos, et al. "A taxonomy and survey of attacks against machine learning." Computer Science Review 34 (2019): 100199.

[19] A Complete List of All (arXiv) Adversarial Example Papers https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html Retrieved: Dec 2022

[20] Deniz, Oscar, et al. "Robustness to adversarial examples can be improved with overfitting." International Journal of Machine Learning and Cybernetics 11.4 (2020): 935-944.

[21] Rice, Leslie, Eric Wong, and Zico Kolter. "Overfitting in adversarially robust deep learning." International Conference on Machine Learning. PMLR, 2020.

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

[23] Gunning, David, et al. "XAI—Explainable artificial intelligence." Science Robotics 4.37 (2019).

[24] Hamon, Ronan, Henrik Junklewitz, and Ignacio Sanchez. "Robustness and explainability of artificial intelligence." Publications Office of the European Union (2020).

[25] Amrani, Moussa, Levi Lúcio, and Adrien Bibal. "ML+ FV= $\heartsuit $? A Survey on the Application of Machine Learning to Formal Verification." arXiv preprint arXiv:1806.03600 (2018).

[26] Hynes, Nick, D. Sculley, and Michael Terry. "The data linter: Lightweight, automated sanity checking for ml data sets." NIPS MLSys Workshop. 2017.

[27] Data Linter https://github.com/brain-research/data-linter Retrieved: Dec, 2022

[28] Data-linter https://pypi.org/project/data-linter/ Retrieved: Dec, 2022

[29] Ng, Andrew. "From Model-centric to Data-centric AI." (2021).

[30] 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.

[31] Bernardi, Lucas, Themistoklis Mavridis, and Pablo Estevez. "150 successful machine learning models: 6 lessons learned at booking. com." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.

[32] Yi, Jeonghee, et al. "Predictive model performance: Offline and online evaluations." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013.

[33] Namiot, Dmitry, et al. "On the applicability and limitations of formal verification of machine learning systems." (2021).

[34] Black swan theory https://en.wikipedia.org/wiki/Black_swan_theory Retrieved: Dec, 2021

[35] Geirhos, Robert, et al. "Shortcut learning in deep neural networks." Nature Machine Intelligence 2.11 (2020): 665-673.

[36] Kaufman, Shachar, et al. "Leakage in data mining: Formulation, detection, and avoidance." ACM Transactions on Knowledge Discovery from Data (TKDD) 6.4 (2012): 1-21.

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

[38] WFCES 2022 https://ide-rus.ru/wfces2022 Retrieved: Dec, 2022