ГОСТ Р 70462.1-2022/ISO/IEC TR 24029-1-2021

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

ИНТЕЛЛЕКТ ИСКУССТВЕННЫЙ

Оценка робастности нейронных сетей

Часть 1

Обзор

Information technology. Artificial intelligence. Assessment of the robustness of neural networks. Part 1. Overview

ОКС 35.020

Дата введения 2023-01-01

Предисловие

1 ПОДГОТОВЛЕН Федеральным государственным автономным образовательным учреждением высшего образования "Национальный исследовательский университет "Высшая школа экономики" (НИУ ВШЭ) на основе собственного перевода на русский язык англоязычной версии документа, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 164 "Искусственный интеллект"

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ

Приказом Федерального агентства по техническому регулированию и метрологии от 2 ноября 2022 г. N 1226-ст

4 Настоящий стандарт идентичен международному документу ISO/IEC TR 24029-1:2021* "Искусственный интеллект (AI). Оценка устойчивости нейронных сетей. Часть 1. Обзор" (ISO/IEC TR 24029-1:2021 "Artificial Intelligence (AI) - Assessment of the robustness of neural networks - Part 1: Overview", IDT).

_____

* Доступ к международным и зарубежным документам, упомянутым в тексте, можно получить, обратившись в Службу поддержки пользователей. - Примечание изготовителя базы данных.

Наименование настоящего стандарта изменено относительно наименования указанного международного документа для приведения в соответствие с

ГОСТ Р 1.5-2012 (пункт 3.5)

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в

статье 26 Федерального закона от 29 июня 2015 г. N 162-ФЗ "О стандартизации в Российской Федерации". Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе "Национальные стандарты", а официальный текст изменений и поправок - в ежемесячном информационном указателе "Национальные стандарты". В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя "Национальные стандарты". Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования - на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.rst.gov.ru)

Введение

При проектировании системы искусственного интеллекта (далее - ИИ) некоторые свойства, такие как робастность, отказоустойчивость, надежность, точность, безопасность, конфиденциальность и т.д., часто считают предпочтительными. Определение робастности приведено в 3.6. Робастность - важнейшее свойство, которое ставит новые задачи в сфере систем ИИ. Например, в рамках управления рисками систем ИИ существуют некоторые риски, которые конкретно связаны с робастностью этих систем и понимание которых имеет важное значение для внедрения ИИ во многих сферах. В настоящем стандарте представлен обзор актуальных подходов для оценки этих рисков с особым упором на нейронные сети, широко использующиеся в промышленности.

В большинстве отраслей промышленности проверка программного обеспечения - это важнейшая часть любого производственного процесса. Задача состоит в том, чтобы обеспечить как безопасность, так и производительность программного обеспечения, используемого во всех частях системы. В некоторых областях процесс верификации программного обеспечения (включая его обновления) также является важной частью сертификации. Например, в автомобильной или авиационной областях при применении действующих стандартов, таких как ИСО 26262 [1] или DO 178C, необходимо предпринимать определенные действия для обоснования дизайна, реализации и тестирования любого встроенного программного обеспечения.

Методы, используемые в системах ИИ, также подлежат валидации. Однако общие методы в системах ИИ создают новые проблемы, которые требуют конкретных подходов для обеспечения адекватного тестирования и/или проверки.

Типы тех систем, которые основаны на технологиях ИИ, включают системы интерполяции/регрессии, классификации, скоринговые и решающие системы.

Хотя существует множество методов валидации систем, не связанных с ИИ, они не всегда непосредственно применимы к системам ИИ и, в частности, к нейронным сетям. Архитектуры нейронных сетей представляют собой особую проблему, поскольку они не поддаются простому анализу и иногда могут быть непредсказуемы ввиду их нелинейной природы, что требует новых подходов к решению возникающих задач.

Методы подразделяются на три группы: статистические методы, формальные методы и эмпирические методы. В настоящем стандарте представлена справочная информация о существующих методах оценки робастности нейронных сетей.

Отмечается, что характеристика робастности нейронных сетей является открытой областью исследований и что существуют ограничения как для подходов тестирования, так и для процессов валидации.

1 Область применения

В настоящем стандарте представлена справочная информация о существующих методах оценки робастности нейронных сетей.

2 Нормативные ссылки

В настоящем стандарте нормативные ссылки не используются.

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями.

ИСО и МЭК поддерживают терминологические базы, используемые в сфере стандартизации и представленные на следующих сайтах:

- платформа онлайн-просмотра ИСО, доступная по адресу: https://www.iso.org/obp;
- Электропедия МЭК, доступная по адресу: https://www.electropedia.org/.

3.1 **искусственный интеллект;** ИИ (artificial intelligence, AI): <Системная> способность спроектированной системы приобретать, обрабатывать и применять знания и навыки.

3.2 **эксплуатационные испытания** (field trial): Испытания новой системы в реальных условиях в соответствии с ее назначением (возможно, с ограниченной группой пользователей).

Примечание - Под условиями понимается как окружающая среда, так и процесс использования.

3.3 **входные данные** (input data): Данные, для которых развертываемая модель машинного обучения вычисляет прогнозируемый результат или вывод.

Примечание - Специалисты по машинному обучению также называют входные данные данными вне выборки, новыми данными и производственными данными.

3.4 **нейронная сеть** (neural network): Сеть примитивных обрабатывающих элементов, соединенных взвешенными связями с регулируемыми весами, в которой каждый элемент выдает значение, применяя нелинейную функцию к своим входным значениям, и передает его другим элементам или представляет его в качестве выходного значения.

Примечания

1 В то время как некоторые нейронные сети предназначены для моделирования функционирования нейронов в нервной системе, большинство нейронных сетей используются в ИИ в качестве реализаций нейросетевой модели.

2 Примерами нелинейных функций являются пороговая функция, сигмоидальная функция и полиномиальная функция.

[ИСО/МЭК 2382:2015, 2120625, изменено - добавлены сокращенные термины, а примечания 3-5 удалены]

3.5 **требование** (requirement): Заявление, которое обозначает или выражает необходимость, а также связанные с ней ограничения и условия.

[ISO/IEC/IEEE 15288:2015, 4.1.37]

3.6 **робастность** (robustness): Способность системы ИИ поддерживать качество работы алгоритмов машинного обучения при любых условиях.

Примечание - В настоящем стандарте главным образом описываются условия, связанные со входными данными, такими как их спектр и характеристики. Но это определение представлено более широко, чтобы не исключать аппаратный сбой и другие виды условий.

3.7 **тестирование** (testing): Деятельность, в которой система или компонент выполняется в определенных условиях, результаты наблюдают или фиксируют, а также проводят оценку какого-либо аспекта системы или компонента.

[ISO/IEC/IEEE 26513:2017, 3.42]

3.8 **тестовые данные** (test data): Подмножество выборок входных данных (см. 3.3), используемых для оценки ошибки обобщения окончательной модели машинного обучения, выбранной из набора возможных моделей машинного обучения [2].

3.9 **обучающие данные** (training dataset): Подмножество выборок, которые подаются в модель машинного обучения.

3.10 **валидация** (validation): Подтверждение посредством предоставления объективных доказательств того, что требования (см. 3.5) для конкретного предполагаемого использования или применения выполнены.

[ИСО/МЭК 25000:2014, 4.41, изменено - примечание 1 удалено]

3.11 **валидационные данные** (validation data): Подмножество выборок входных данных (3.3), используемых для оценки ошибки прогнозирования возможной модели машинного обучения [2].

Примечание - Валидация (3.10) модели машинного обучения может быть использована для выбора модели машинного обучения.

3.12 **верификация** (verification): Подтверждение посредством предоставления объективных доказательств того, что указанные требования выполнены.

[ИСО/МЭК 25000:2014, 4.43, изменено - примечание 1 удалено]

4 Обзор существующих методов оценки робастности нейронных сетей

4.1 Общие положения

4.1.1 Концепция робастности

Цели обеспечения робастности направлены на то, чтобы ответить на вопросы "Какая степень робастности требуется системе?" или "Какие свойства

робастности представляют интерес?". Свойства робастности показывают, насколько четко система обрабатывает новые данные по сравнению с результатами обработки данных, ожидаемых в типовых операциях.

4.1.2 Типичный рабочий процесс для оценки робастности

В настоящем пункте рассмотрено проведение оценки робастности нейронных сетей в различных задачах ИИ, таких как классификация, интерполяция и другие сложные задачи.

Существуют различные способы оценки робастности нейронных сетей с использованием объективной информации. Типичный рабочий процесс для определения робастности нейронной сети (или другого метода) представлен на рисунке 1.

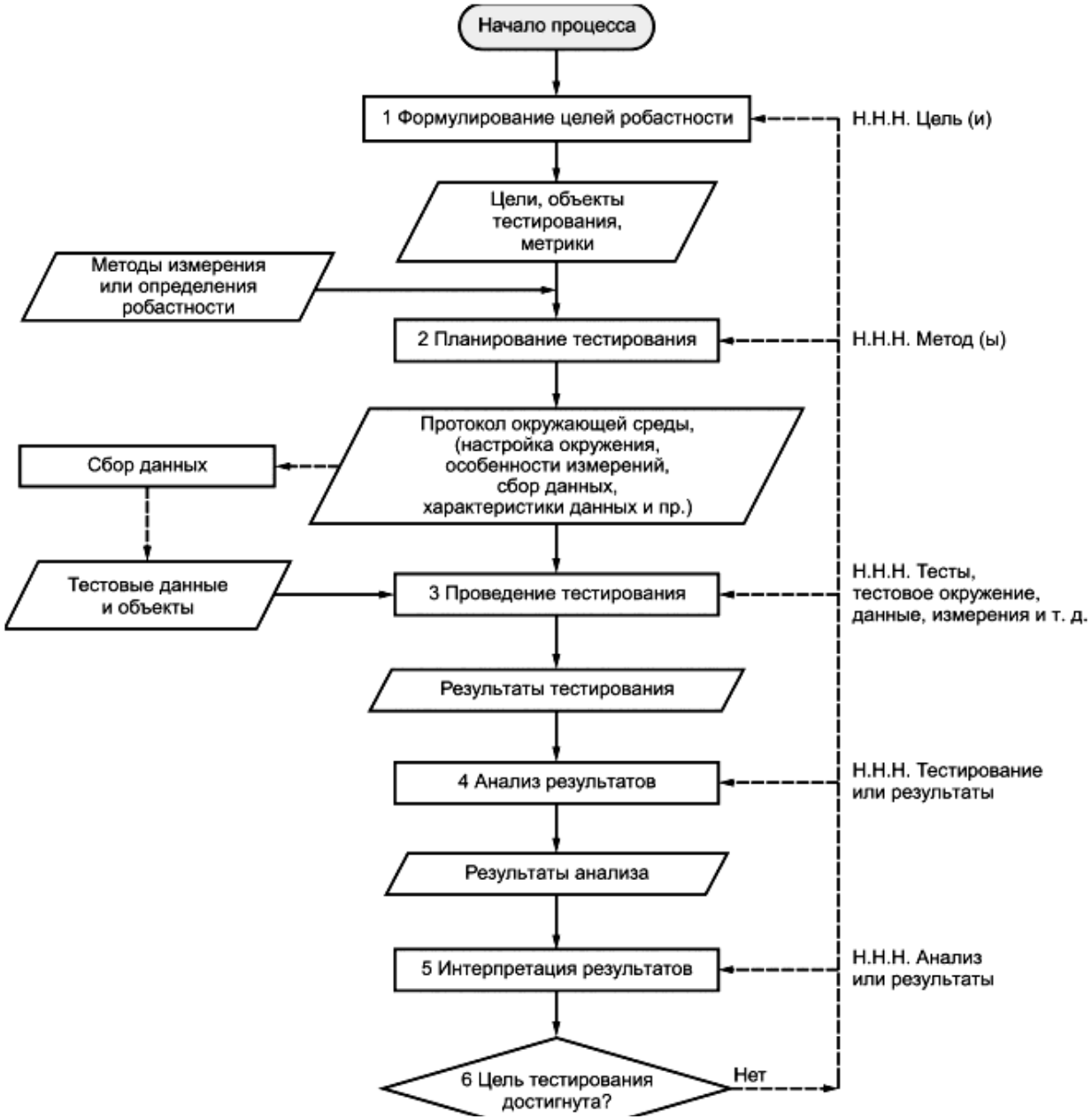
Начало процесса
1 Формулирование целей робастности
Н.Н.Н. Цель (и)
Цели, объекты тестирования, метрики
Методы измерения или определения робастности
2 Планирование тестирования
Н.Н.Н. Метод (ы)
Протокол окружающей среды, (настройка окружения, особенности измерений, сбор данных, характеристики данных и пр.)
Сбор данных
Тестовые данные и объекты
3 Проведение тестирования
Н.Н.Н. Тесты, тестовое окружение, данные, измерения и т. д.
Результаты тестирования
4 Анализ результатов
Н.Н.Н. Тестирование или результаты
Результаты анализа
5 Интерпретация результатов
Н.Н.Н. Анализ или результаты
6 Цель тестирования достигнута?
Нет

Рисунок 1 - Типичный рабочий процесс для определения робастности нейронной сети

Шаг 1 Формулировка целей робастности

Процесс начинается с формулирования целей обеспечения робастности. На начальном этапе должны быть идентифицированы объекты тестирования, подлежащие верификации на робастность. С их учетом впоследствии определяют количественные метрики оценки тех элементов, которые показывают достижение робастности. Все это образует набор критериев для принятия решений о свойствах робастности, которые могут быть предметом дальнейшего утверждения соответствующими заинтересованными сторонами (см. ISO/IEC/IEEE 16085:2021, 7.4.2, [3]).

Шаг 2 Планирование тестирования

Этот шаг заключается в планировании проверок, которые демонстрируют робастность. Эти проверки опираются на различные методы, например статистические, формальные или эмпирические. На практике используется комбинация методов. Статистические подходы обычно опираются на процесс математического тестирования и способны проиллюстрировать определенный уровень достоверности результатов. Формальные методы полагаются на формальные доказательства для демонстрации математических свойств в области определения модели. Эмпирические методы основаны на экспериментировании, наблюдении и экспертной оценке. При планировании проверки необходимо определение настроек среды, планирование сбора данных и определение характеристик данных (какие типы данных в каких диапазонах будут использованы, какие граничные условия будут нарушены для проверки робастности и т.д.). Результатом шага 2 является протокол тестирования, который представляет собой документ, выражающий смысл, цели, дизайн и предлагаемый анализ, методологию, мониторинг, проведение тестирования, а также хранение его результатов (более детально содержание протокола тестирования доступно в определении плана клинического исследования, изложенного в ИСО 14155:2020, 3.9, [4]).

Шаг 3 Проведение тестирования

Далее проводят тестирование согласно составленному протоколу тестирования и сбор результатов. Допускается выполнение тестов с использованием реальной среды или моделирования (симуляции) реальной среды, а также потенциально путем комбинации этих двух подходов.

Шаг 4 Анализ результатов

После завершения тестирования результаты тестов анализируют с использованием метрик, выбранных на шаге 1.

Шаг 5 Интерпретация результата

Результаты анализа интерпретируют для принятия обоснованного решения.

Шаг 6 Цель тестирования достигнута?

Решение по робастности системы формулируют по определенным ранее критериям и полученной интерпретации результатов анализа.

Если цели тестирования не достигнуты, проводят анализ процесса, и процесс возвращается к соответствующему предшествующему шагу с целью устранить недостатки, например: путем добавления целей робастности, модификации или добавления метрик, учета различных аспектов для измерения, перепланирования тестов и т.д.

Системы ИИ, которые в значительной степени полагаются на нейронные сети, особенно глубокие нейронные сети (deep neural networks, DNN), имеют недостатки, которые проявляются в виде сбоев поведения системы, напоминающих аналогичные эффекты в программном обеспечении. Типичные ситуации продемонстрированы путем подачи "неблагоприятных примеров" в системы распознавания объектов, например [5]. Эти встроенные ошибки DNN "исправить" непросто. Исследования по этой проблеме показывают, что существуют меры для повышения устойчивости DNN к неблагоприятным примерам, но это работает до определенной степени [6], [7]. Однако, если дефект обнаружен во время процедуры тестирования, система ИИ может сигнализировать о проблеме при обнаружении соответствующего шаблона ввода.

Сбор данных

Сбор данных представляет собой процесс выбора, создания и/или генерации тестовых данных и объектов, необходимых для проведения тестирования.

Иногда этот процесс включает в себя рассмотрение юридических или других нормативных требований, а также различных практических или технических вопросов.

Протокол тестирования содержит требования и критерии, необходимые для сбора данных. Проблемы и методы сбора данных не рассматриваются детально в настоящем стандарте.

Значительное влияние на робастность могут оказывать следующие факторы:
- масштаб значений отдельных измерений;
- разнообразие, репрезентативность и диапазон выбросов;
- выбор реальных или синтетических данных;
- наборы данных, специально используемые для тестирования робастности;
- состязательные и другие примеры, которые исследуют гипотетические крайности предметной области;
- состав наборов данных для обучения, тестирования и валидации.

4.2 Классификация методов

Следуя описанному выше рабочему процессу определения робастности, в настоящем стандарте представлены методы и метрики, применимые к различным типам тестирования, то есть статистические, формальные и эмпирические методы.

Статистические подходы, как правило, основаны на математической оценке некоторых наборов данных, способствуя обеспечению определенного уровня достоверности результатов. Формальные методы полагаются на надежное формальное доказательство, чтобы продемонстрировать математическое свойство в предметной области. В настоящем стандарте формальные методы не ограничены областью синтаксической теории доказательств и включают методы проверки корректности, такие как проверка модели. Эмпирические методы базируются на экспериментах, наблюдениях и экспертных оценках.

Несмотря на то что систему можно охарактеризовать посредством наблюдения или доказательства, в настоящем стандарте выбрано разделение методов наблюдения на статистические и эмпирические. Статистические методы генерируют воспроизводимые показатели робастности на основе заданных наборов данных. Эмпирические методы формируют данные, которые можно проанализировать статистическими методами, но они не обязательно воспроизводимы из-за включения субъективной оценки. Поэтому необходимо, чтобы методы из обеих категорий применялись совместно.

Таким образом, в настоящем стандарте сначала рассмотрены статистические подходы, которые являются наиболее распространенными при оценке робастности. Для них характерен подход к тестированию, определяемый методологией с использованием математических метрик. Затем исследованы подходы к получению формального доказательства, которые используют для оценки робастности и, наконец, представлены эмпирические подходы, основанные на субъективных наблюдениях, которые дополняют оценку робастности, когда статистические и формальные подходы недостаточны или нецелесообразны.

Эти методы не используют для прямой оценки робастности в целом. Каждый из них нацелен на различные аспекты робастности, предоставляя несколько частичных показателей, сочетание которых позволяет оценить робастность.

Эксперты по оценке робастности используют эти методы, чтобы ответить на различные виды вопросов по системе, которую они проверяют, например:

- статистические методы позволяют эксперту по оценке проверить, достигают ли свойства систем предпочтительного целевого порога (например, сколько дефектных единиц произведено);

- формальные методы позволяют эксперту по оценке проверить, доказуемы ли свойства в области использования (например, всегда ли система работает в заданных пределах безопасности);

- эмпирические методы позволяют эксперту оценить ту степень, в которой свойства системы достоверны в тестируемом сценарии (например, является ли наблюдаемое поведение удовлетворительным).

Принцип применения таких методов к оценке робастности заключается в том, чтобы оценить, в какой степени эти свойства сохраняются при изменении условий:
- при использовании статистических методов: как изменение условий влияет на измеренные рабочие характеристики?
- в случае формальных методов: сохраняются ли необходимые свойства при расширении границ области условий (входных данных)?
- при применении эмпирических методов: сохраняются ли свойства в других сценариях?

Следует отметить, что характеристика робастности нейронных сетей является активной областью исследований, и существуют ограничения как для подходов к тестированию, так и к валидации. При использовании подходов к тестированию вариация возможных входных данных вряд ли будет достаточно большой, чтобы обеспечить какие-либо гарантии эффективности системы. Валидационные подходы обычно требуют аппроксимаций для обработки входных данных высокой размерности и большого количества параметров нейронной сети.

5 Статистические методы

5.1 Общие положения

Одним из аспектов робастности является влияние меняющегося окружения на количественные характеристики входных данных, для анализа которых особенно подходят статистические методы. Эти методы позволяют проводить прямую оценку эффективности в различных сценариях с использованием сравнительного анализа.

При использовании статистических методов для вычисления робастности применяют четыре основных критерия:

1) Подходящие оценочные данные. Для того чтобы оценить робастность модели, сначала устанавливают статистические характеристики распределения данных и определяют набор данных, который охватывает входные условия для целевого приложения, либо посредством сбора реальных данных измерений, либо смоделированных данных. Возможны несколько источников данных, таких как: зашумленные данные, которые не учтены при первоначальном обучении модели; данные из приложений аналогичной предметной области; данные из другого, но эквивалентного источника данных. Хотя общего метода оценки релевантности набора данных не существует, и он часто основан на суждениях человека, существуют некоторые методы (например, основанные на промежуточных представлениях данных) для поддержки этого анализа с помощью различных показателей. Оценка робастности моделей нейронных сетей может меняться при использовании различных наборов тестовых данных.

2) Выбор настройки модели. Оценка позволяет сделать заключение о робастности с использованием различных настроек обученной модели (например, точность модели, квантованный вес и т.д.).

3) Выбор метрики или метрик эффективности. В зависимости от контекста, поставленной задачи и характера данных некоторые метрики не всегда могут быть подходящими, поскольку они могут привести к недостоверным результатам. Надлежащий набор метрик (см. 5.2) помогает избежать подобных ситуаций.

4) Метод принятия решения о робастности. Учитывая выбранную метрику, выполняют статистический тест для принятия решения относительно того, является ли модель робастной.

Свойство робастности, оцениваемое с помощью статистических методов, определяется одним или несколькими пороговыми значениями по набору метрик, которые должны быть выполнены на некоторых тестовых данных. Оценка робастности зависит от конкретного случая, учитывая, что определенные организации или ситуации потребуют других целей и метрик робастности, чтобы определить, достигнута ли цель.

Настоящий подраздел соответствует общему рабочему процессу оценки робастности нейронной сети, представленному на рисунке 1. В частности, он сфокусирован на шагах 1, 2 и 3 рабочего процесса, определенного в 4.1.2, а именно на формулировке целей робастности, планировании тестирования и проведении тестирования.

В 5.2 и 5.3 представлены метрики и методы для статистической оценки робастности нейронной сети, более подробная информация по которым доступна в [8], [9], [10] и [11].

5.2 Метрики робастности, имеющиеся в распоряжении статистических методов

5.2.1 Общие положения

В настоящем пункте представлена справочная информация о доступных статистических показателях, которые обычно применяют к выходу нейронных сетей. Здесь приведено описание целей робастности с использованием шага 1 на рисунке 1. Цели робастности должны быть четко определены. Например, простая формулировка, такая как "обученная нейронная сеть должна быть робастной к входным данным, отличным от тех, на которых она была обучена", является недостаточно четко определенной. В зависимости от входных данных нейронная сеть может полностью соответствовать или вовсе не соответствовать этой целевой функции. С одной стороны, нейронная сеть может быть полностью робастной к входным данным, которые следуют распределению, отличному от исходных обучающей и тестовой выборок, но остаются в пределах области определения. С другой стороны, вполне возможна нейронная сеть, которая вообще не соответствует требованиям, если входные данные находятся в совершенно другой области определения, чем те, на которых она была обучена.

Следовательно, целевая функция робастности должна быть сформулирована в достаточной степени, чтобы можно было определить робастность нейронной сети.

Пример четко поставленной цели (структурированной из трех частей) выглядит следующим образом:

- нейронная сеть должна быть устойчивой к входным данным, отличным от тех, на которых она была обучена;
- предполагается, что входные данные относятся к одной области и могут включать как физически реализуемые, так и гипотетические;
- показатели, которые могут быть использованы, включены в 5.2.2.

В зависимости от задачи, решаемой системой ИИ (например, классификация, интерполяция/регрессия), возможны различные статистические метрики. В настоящем подразделе описаны общие статистические метрики и способ их вычисления. Список не является исчерпывающим, и некоторые из этих показателей совместимы с другими задачами. Их можно использовать как отдельно, так и в комбинации. В зависимости от применения существует также множество метрик, специфичных для конкретной задачи [например, BLEU, TER или METEOR для машинного перевода, отношение пересечений и объединений (intersection over union) для обнаружения объектов на изображениях или средняя точность (mean average precision) для качественного ранжированного поиска], но их описание выходит за рамки настоящего стандарта.

5.2.2 Примеры метрик эффективности для интерполяции

5.2.2.1 Среднеквадратичная ошибка или среднеквадратичное отклонение

Среднеквадратичная ошибка (RMSE) - это стандартное отклонение остатков (ошибок прогнозирования). Ошибки прогнозирования - это показатель того, насколько далеко от линии регрессии находятся точки данных, а RMSE - это показатель разброса остатков.

5.2.2.2 Максимальная ошибка

Максимальная ошибка (max error) - это абсолютная или относительная метрика, вычисляющая значение в исходных данных и соответствующее значение в прогнозе системы ИИ. Абсолютная максимальная ошибка - это максимальная разность между значением в исходных данных и соответствующим значением в прогнозе системы ИИ. Относительная максимальная ошибка - это отношение абсолютной максимальной ошибки к реально измеренному значению.

5.2.2.3 Фактическая и прогнозируемая корреляции

Фактическая/прогнозируемая корреляция (actual/predicted correlation) - это линейная корреляция (в статистическом смысле) между фактическими значениями и прогнозируемыми значениями для каждого значения, рассматриваемого в наборе.

5.2.3 Примеры показателей эффективности для классификации

5.2.3.1 Общие понятия и связанные с ними базовые метрики

Набор образцов может иметь следующие характеристики:

- общая совокупность (total population): общее количество образцов в данных;
- положительные образцы (condition positive, CP): количество реальных положительных образцов в данных;
- отрицательные образцы (condition negative, CN): количество реальных отрицательных образцов в данных;
- положительный прогноз (prediction positive, PP): количество образцов, классифицированных как положительные;
- отрицательный прогноз (prediction negative, PN): количество образцов, классифицированных как отрицательные;
- распространенность (prevalence): доля определенного класса в общем количестве образцов.

Каждый экземпляр в наборе образцов классифицируется системой классификации по одному из следующих принципов:

- истинно положительный экземпляр (TP, попадание): экземпляр принадлежит классу и прогнозируется как принадлежащий классу;
- истинно отрицательный экземпляр (TN, правильный отказ): экземпляр не принадлежит классу и прогнозируется как не принадлежащий классу;
- ложноположительный экземпляр (FP, ложная тревога, ошибка типа I): экземпляр не принадлежит классу и прогнозируется как принадлежащий классу;
- ложноотрицательный экземпляр (FN, промах, ошибка типа II): экземпляр принадлежит классу и прогнозируется как не относящийся к классу.

Несколько метрик построены на основе этих выборочных характеристик, как представлено в таблице 1:

- доля истинно положительных результатов (true positive rate, TPR), чувствительность (sensitivity): доля истинно положительных результатов (также известная как чувствительность, полнота или вероятность обнаружения) указывает на долю объектов, правильно классифицированных как положительные, в общем количестве действительно положительных объектов;

- доля истинно отрицательных результатов (true negative rate, TNR), специфичность (specificity): доля истинно отрицательных результатов (также известная как специфичность или избирательность) указывает долю объектов, правильно классифицированных как отрицательные, в общем количестве отрицательных объектов;

- доля ложноположительных результатов (false positive rate, FPR): доля ложноположительных результатов (также известная как выпадение или вероятность ложной тревоги) указывает долю объектов, ошибочно классифицированных как положительные, которые являются отрицательными. Таким образом задается вероятность ложной тревоги;

- доля ложноотрицательных результатов (false negative rate, FNR): доля ложноотрицательных результатов (также известная как доля промахов) указывает на долю объектов, ложно классифицированных как отрицательные, в общем количестве положительных объектов;

- достоверность (accuracy, ACC): достоверность указывает долю всех правильно классифицированных объектов;

- положительная прогностическая ценность (positive predictive value, PPV): положительная прогностическая ценность (также известная как точность или релевантность) указывает долю результатов, правильно классифицированных как положительные среди общего числа результатов, классифицированных как положительные;

- отрицательная прогностическая ценность (negative predictive value, NPV): отрицательная прогностическая ценность (также известная как способность разделения) указывает долю результатов, правильно классифицированных как отрицательные среди общего числа результатов, классифицированных как отрицательные;

- коэффициент ложного обнаружения (false discovery rate, FDR): коэффициент ложного обнаружения указывает соотношение ошибочно отклоненных нулевых гипотез (ложные срабатывания, ложные тревоги, ошибки типа I) к общему количеству отклоненных нулевых гипотез (положительные результаты прогнозирования);

- коэффициент ложных пропусков (false omission rate, FOR): коэффициент ложных пропусков указывает на соотношение ошибочно отклоненных ложных отрицательных результатов к общему количеству прогнозируемых отрицательных результатов;

- отношение положительного правдоподобия R_{L+} (likelihood relation R_{L+}): положительное отношение правдоподобия указывает отношение истинных положительных результатов к количеству ложноположительных результатов;

- отношение отрицательного правдоподобия R_{L-} (likelihood relation R_{L-}): отношение отрицательного правдоподобия указывает отношение ложноотрицательных результатов к количеству истинно отрицательных результатов;

- диагностическая вероятность (diagnostic odds rate, DOR): указывает отношение вероятности истинных положительных результатов к вероятности ложных положительных результатов и не зависит от распространенности.

Таблица 1 - Характеристики выборки и соответствующие базовые показатели, построенные на их основе

		Истинные			
	Вся популяция, общее количество	Положительные образцы CP	Отрицательные образцы CN	Распространенность $\frac{N_{C+}}{P_{tot}}$	Достоверность $\frac{N_{T+}+N_{T-}}{P_{tot}}$
Предсказанные	Положительные образцы	Истинно положительные экземпляры TP Мощность	Ложноположительные экземпляры FP Ошибка I рода	Положительная прогностическая ценность V_{P+} Точность, релевантность $\frac{N_{T+}}{N_{P+}}$	Доля ложных открытий $\frac{N_{F+}}{N_{P+}}$

<table>
<tr><td></td><td>Отрицательные образцы</td><td>Ложноотрицательные экземпляры FN

Ошибка II рода</td><td>Истинно отрицательные экземпляры TN</td><td>Коэффициент ложных пропусков
$\frac{N_{F-}}{N_{P-}}$</td><td colspan="2">Отрицательная прогностическая ценность
$\frac{N_{T-}}{N_{P-}}$</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>Доля истинно положительных результатов R_{T+}

Чувствительность, полнота

Вероятность определения
$\frac{N_{T+}}{N_{C+}}$</td><td>Доля ложноположительных результатов R_{F+}

fall-out, вероятность ложной тревоги
$\frac{N_{F+}}{N_{C-}}$</td><td>Отношение положительного правдоподобия
R_{L+}

$\frac{R_{T+}}{R_{F+}}$</td><td rowspan="2">Диагностическая вероятность
$\frac{R_{L+}}{R_{L-}}$</td><td rowspan="2">Оценка F1

$\left(\frac{R_{T+}^{-1}+V_{P+}^{-1}}{2}\right)^{-1}$</td></tr>
<tr><td>Доля ложно-отрицательных результатов R_{F-}

Доля промахов,
$\frac{N_{F-}}{N_{C+}}$</td><td>Доля истинно отрицательных результатов R_{T-}

Специфичность, избирательность
$\frac{N_{T-}}{N_{C-}}$</td><td>Отношение отрицательного правдоподобия
R_{L-}

$\frac{R_{F-}}{R_{T-}}$</td></tr>
</table>

где N_{T+} - количество истинных положительных результатов;

N_{T-} - количество истинных отрицательных значений;

N_{F+} - количество ложноположительных результатов;

N_{F-} - количество ложноотрицательных результатов;

N_{C+} - число положительных условий;

N_{C-} - число отрицательных условий;

P_{tot} - общее количество наблюдений;

N_{P+} - число положительных прогнозов;

N_{P-} - число отрицательных прогнозов;

R_{T+} - доля истинно положительных результатов;

R_{T-} - доля истинно отрицательных результатов;

R_{F+} - доля ложноположительных результатов;

R_{F-} - доля ложноотрицательных результатов;

R_{L+} - отношение положительного правдоподобия;

R_{L-} - отношение отрицательного правдоподобия;

V_{P+} - величина положительной прогностической ценности.

В таблице 1 представлено обобщенное представление характеристик и показателей выборки, описанных в настоящем подразделе. Все эти выборочные характеристики и метрики применимы в первую очередь к бинарной классификации, но также имеют обобщенные определения в многоклассовых случаях и случаях со множественными метками.

5.2.3.2 Расширенные метрики

Кривая точности-полноты

Пары метрик "точность/полнота" вычисляют при разных пороговых значениях вывода. Пары "точность/полнота" отражают компромиссы между точностью и полнотой, когда эти метрики используют для оценки робастности.

Рабочая характеристика приемника (ROC)

Кривая ROC (Receiver operating characteristic) представляет собой график зависимости пропорции истинно положительных результатов в зависимости от пропорции ложноположительных результатов при различных настройках гиперпараметров (например, порога принятия решения).

ROC отражает компромисс между долями истинно положительных и ложноположительных показателей, когда эти показатели используют для оценки робастности. Кривые ROC применяют, когда один показатель связан со значительными затратами или преимуществами при оценке робастности, например: в области медицины, где ложные диагнозы могут приводить к критическим последствиям.

5.2.3.3 Подъем (lift)

Метрика подъема - это мера, сравнивающая относительную эффективность системы прогнозирования с другой контрольной группой (обычно выбираемой случайным образом).

5.2.3.4 Площадь под кривой

Площадь под кривой измеряет интеграл кривой рабочих характеристик приемника ROC, которая представляет эффективность модели для каждого порога классификации. Кривая ROC показывает долю истинных положительных результатов относительно доли ложноположительных результатов.

5.2.3.5 Сбалансированная достоверность

Сбалансированная достоверность (balanced accuracy) - это средняя полнота, полученная по каждому классу [12].

5.2.3.6 Микроусреднение и макроусреднение

В случаях несбалансированных наборов данных такие показатели, как точность или полнота, рассчитанные для всего набора данных, иногда дезориентируют. Возможной стратегией для решения этой проблемы является вычисление метрики макроусреднения, которая представляет собой среднее значение показателя, вычисленного для каждого класса отдельно, вместо метрики микроусреднения, которую используют стандартным вычислением без разделения классов [13].

5.2.3.7 Коэффициент корреляции Мэтьюза

Коэффициент корреляции Мэтьюза (Matthews correlation coefficient, MCC) - это мера по набору классификаций (предсказаний). Его диапазон лежит в пределах [-1, +1], в котором +1 представляет точное предсказание, -1 - противоположное предсказание, а 0 - среднее предсказание. Следует отметить, что эта метрика

обобщается в тех случаях, когда классы не сбалансированы в исходных данных (то есть значение MCC равно 0 для случайного классификатора на *N* классах, даже если точность этого классификатора отличается от 1/*N*) [14], [15].

Коэффициент корреляции Мэтьюза MCC вычисляют по формуле

$$\mathrm{MCC} = \frac{N_{T+} \cdot N_{T-} - N_{F+} \cdot N_{F-}}{\sqrt{(N_{T+} + N_{F+})(N_{T+} + N_{F-})(N_{T-} + N_{F+})(N_{T-} + N_{F-})}}, \quad (1)$$

где N_{T+} - количество истинных положительных результатов;

N_{T-} - количество истинных отрицательных значений;

N_{F+} - количество ложноположительных результатов;

N_{F-} - количество ложноотрицательных результатов.

5.2.3.8 Матрица ошибок и связанные метрики

Матрица ошибок (confusion matrix) позволяет провести подробный анализ эффективности классификатора и помочь обойти или выявить слабые места отдельных метрик, поскольку она обеспечивает более четкий и всесторонний анализ эффективности классификатора. Напротив, использование матрицы ошибок в качестве единственной меры эффективности классификатора недостаточно информативно для проведения этого анализа, так как оно не указывает, какие классы наиболее распознаются или какой тип ошибок совершает классификатор.

Матрица ошибок *C* представляет собой квадратную матрицу, где запись $C_{r,c}$ в строке *r* и столбце *c* - это количество экземпляров, принадлежащих к классу или категории *r*, которые классифицируют как принадлежащие к классу *c*.

Матрицы ошибок включают количество истинно положительных, истинно отрицательных, ложноположительных и ложноотрицательных результатов: на их основе можно рассчитать такие метрики, как достоверность, полнота по классам и точность. Из элементов матрицы ошибок могут быть получены дополнительные метрики, такие как энтропия гистограммы, представленная матрицей.

5.2.4 Другие меры

5.2.4.1 Кусочно-линейная функция потерь

Кусочно-линейная функция потерь (hinge loss) - верхняя граница количества ошибок, сделанных классификатором. В общем случае для классификации с несколькими классами дистанцию до границы вычисляют методом Краммера-Зингера [16].

5.2.4.2 Каппа Коэна

Каппа Коэна - это мера согласия между экспертами, выполняющими такую же задачу, как и оцениваемая система κ, вычисляемая по формуле

$$\kappa = (p_o - p_e)/(1 - p_e), \quad (2)$$

где p_o - априорная вероятность согласованности меток на любой выборке в наблюдаемых данных;

p_e - ожидаемое согласие, когда каждый из двух экспертов присваивает метки независимо и в соответствии с собственными измеренными априорными распределениями с учетом эмпирических данных.

Эта мера полезна, когда не обязательно существует золотой стандарт оценки, например когда метки, предоставленные человеком, также являются неточными или когда таких меток не существует, и для сравнения доступны только автоматизированные методы.

В основном эту меру используют для оценки качества данных после сделанных человеком аннотаций (подверженных ошибкам), но ее также применяют в качестве вспомогательного метода оценки, когда метки отсутствуют, путем сравнения двух классификаторов друг с другом.

5.3 Статистические методы измерения робастности нейронной сети

5.3.1 Общие положения

При применении метрик по 5.2 к тестовым данным для оценки робастности доступно несколько статистических методов. В этом подразделе описаны некоторые из доступных статистических методологий для выполнения шагов 2 и 3, представленных в 4.1, для планирования и проведения тестирования. Выполнение протокола тестирования не является уникальным для нейронных сетей, и подготовка включает настройку тестового окружения, сведения о том, что и как измерять, а также сбор данных и прочие характеристики. Разница в планировании тестирования робастности нейронных сетей заключается в необходимости более тщательного сбора данных (например, об уровне качества, степени детализации, наборах данных для обучения/тестирования/валидации и т.д.). При проведении тестирования источник данных и доступность вычислительных ресурсов являются существенными вследствие того, что нейронные сети требуют в некоторых случаях значительных объемов данных и вычислительных ресурсов.

5.3.2 Контрастные меры

Статистические показатели эффективности применяют сначала к базовому набору данных, а затем к одному или нескольким наборам данных, отражающим целевые изменения условий. Если для каждого из них снижение производительности по сравнению с эталонным тестовым набором достаточно низкое, то систему считают надежной.

6 Формальные методы

6.1 Общие положения

Другим аспектом робастности является степень, в которой изменяющиеся обстоятельства влияют на поведение системы независимо от ее эффективности. Формальные методы подходят для оценки стабильности системы, т.е. степени, в которой ее результат изменяется при изменении входных данных. Хотя робастная система может быть нестабильной, а стабильная система может быть неробастной, стабильность является надежным показателем робастности, поскольку она делает результат более предсказуемым.

Формальные методы использовались для повышения надежности программного обеспечения и обеспечения более строгого контроля качества систем, включающих программное обеспечение. Хотя формальные методы в основном применялись в контексте приложений, критически важных для безопасности, например в транспортных системах, в настоящее время они используются более широко.

Формальные методы позволяют получить математическое доказательство свойства по всей данной области, тогда как статистические или эмпирические методологии основаны на экстраполяции результатов только из проверенных выборок на всю область. Свойства безопасности обычно находятся в центре внимания, но эти методы применимы к широкому спектру свойств. Формальные методы, как правило, сложны в применении, поскольку они иногда требуют конкретного математического моделирования в зависимости от типа анализируемой системы и, возможно, более сложного инструментального оснащения (instrumentation) системы.

Программные системы ИИ создают новые специфические проблемы с точки зрения валидации системы, например: в отличие от классического программного обеспечения их поведение сложнее объяснить и доказать. Это особенно характерно для нейронных сетей из-за способа их построения (посредством обучения вместо программирования) и присущей нелинейности их поведения. В результате требуется найти более подходящие системные свойства, демонстрирующие робастность системы ИИ, и рассматриваются новые методологии, доказывающие их достоверность.

Настоящий раздел следует общему рабочему процессу для оценки робастности нейронной сети, изображенному на рисунке 1. В частности, он сосредоточен на шагах 1, 2 и 3, представленных в 4.1 и состоящих из выбора требуемых свойств робастности, затем - из подготовки данных, а далее - их выполнения тестов. В зависимости от видов использования нейронной сети следует учитывать различные свойства, иногда необходимы различные этапы подготовки и возможны несколько методов для проведения тестирования.

6.2 Цель робастности, достижимая с использованием формальных методов

6.2.1 Общие положения

В настоящем подразделе описано свойство стабильности, которое является одним из свойств, используемых для оценки робастности. В зависимости от задачи, выполняемой системой, оно формализовано следующими путями:

1) для систем интерполяции: стабильность интерполяции вычисляет неопределенность нейронной сети, что позволяет определить, когда нейронная сеть может иметь недостаточную робастность;

2) для систем классификации: максимальное стабильное пространство вычисляет размер той области, в которой нейронная сеть будет иметь стабильную эффективность классификации.

6.2.2 Стабильность интерполяции (Interpolation stability)

В некоторых случаях нейронные сети используют для замены сложных математических вычислений, выполнение которых требует больших затрат. Например, для сложной системы дифференциальных уравнений, требующей решения итерационным методом (например, подход Ньютона-Рафсона), нейронную сеть применяют в качестве "оракула", который выводит определенные удовлетворяющие условия систем. Для реализации такого метода набор данных для обучения, тестирования и валидации генерируется заранее (в идеале - однократно), но, если качество набора данных недостаточно для обучения нейросети нужной точности предсказания, этап генерации может повторяться. Вопрос о релевантности и полноте покрытия пространства исходных параметров имеет важное значение для определения качества поведения нейронной сети. Хотя очевидно, что обучающий набор данных не является исчерпывающим (не охватывает всего пространства входов/выходов), проблема остается той же для набора тестовых данных, который используют для проверки. Кроме того, поскольку системы интерполяции обычно применяют для моделирования функций, работающих более чем в двух измерениях, нейронные сети создают сложности как сточки зрения поведения, так и визуализации.

Для систем интерполяции главным преимуществом нейронных сетей является их способность эффективно моделировать сложное линейное и нелинейное поведение. Однако по своей природе в какой-то момент возникает некоторое нелинейное поведение, что позволяет нейронной сети демонстрировать непредсказуемое поведение в тех частях области, в которых она должна быть использована. Одним из следствий этого является то, что в одних регионах области поведение нейросетей может быть неустойчивым, а в других - устойчивым. Поскольку сложно охватить каждый регион области, чтобы найти эти неустойчивые поведения, то весьма вероятно, что такие области будут упущены, и это снизит робастность. Такое незапланированное поведение составляет неопределенность в оценке качества интерполяционной способности нейронной сети.

6.2.3 Максимальное стабильное пространство для сопротивления возмущениям

Для снижения рисков, связанных с возмущениями данных, которые также называются неблагоприятными (adversarial) примерами, можно доказать устойчивость классификатора до определенного момента. Для достижения этой цели используют свойство максимального стабильного пространства. Считают, что для конкретного входного сигнала отсутствует неблагоприятный пример. Понятие расстояния необходимо для определения набора входных данных, которые находятся "вокруг" конкретной точки. В приложении А приведено описание нескольких типов искажений данных, а также соответствующих метрик расстояния для оценки нейронных сетей.

В настоящем пункте описаны три общих подхода, которые могут продемонстрировать свойство максимального стабильного пространства, используемое для измерения робастности нейронной сети, выполняющей классификацию. Это свойство может быть доказано следующими способами:

1) как логическая задача, разрешаемая с использованием решателя;

2) как числовая проблема, решаемая с помощью алгоритма оптимизации, посредством которого вычисляют максимум;

3) как математическое свойство, которое аппроксимируется путем абстрактной интерпретации.

Каждый метод был адаптирован с учетом специфики нейронных сетей.

6.3 Проведение тестирования формальными методами

6.3.1 Использование анализа неопределенности для доказательства стабильности интерполяции

Анализ неопределенности - это метод, обычно используемый для управления поведением математических функций. Цель состоит в том, чтобы определить, на каких входах функция имеет внезапные и существенные изменения. Для нейронных сетей - это действенный способ обнаружить проблемы, описанные в 6.2.2, в частности: можно сравнить измеренное поведение с представлением о фактическом поведении. Цель состоит в том, чтобы измерить способность нейронной сети моделировать в пределах допустимого диапазона отклонений явление, которое она предназначена моделировать. Это способствует измерению вариации отклика

сети, чтобы убедиться в отсутствии нестабильного поведения.

Проделана фундаментальная работа по формализации стабильности нейронной сети. Например, в [17] описан метод расчета распространения неопределенности в сети, позволяющий таким образом обнаруживать регион, в котором ответ нейронной сети является ненормальным, а неробастное поведение должно быть ожидаемым. В [18] представлен метод, показывающий влияние или важность входных переменных на выход, независимо от природы переменных (непрерывная или дискретная). В [19] приведено несколько источников неопределенности, включая неопределенность входных данных, чувствительность сети и влияние случайности разбиения наборов данных для обучения и тестирования. Таким образом, можно определить условия, при которых сеть не является робастной, и установить причину неопределенности в отклике сети.

6.3.2 Использование решателя для доказательства свойства максимального стабильного пространства

Известно, что нейронные сети, как правило, достаточно большие, нелинейные, невыпуклые и недоступны для универсальных инструментов, таких как решатели линейного программирования или существующие теории выполнимости по модулю (satisfiability modulo theories, SMT). Тем не менее получено несколько достижений при использовании технологий решателей для подтверждения свойств в нейронных сетях. В качестве примера в [20] представлен подход к эффективному доказательству свойств над некоторыми классами нейронных сетей [с использованием функций активации ReLU (Rectified Linear Unit), определенной как ReLU(x) = max(0, x)] посредством варианта симплекс-алгоритма. В [21] решатель SMT применяют для доказательства отсутствия или существования неблагоприятного примера, включая возможность его демонстрации. В [22] рассмотрена комбинация решения выполнимости и линейного программирования на линейной аппроксимации общего поведения сети. В этих работах отражена возможность адаптировать общие технологии решателей для подтверждения свойств в нейронных сетях, доказать робастность классификаторов, а также использовать данные методы для решения других задач нейронных сетей.

6.3.3 Использование методов оптимизации для доказательства свойства максимально стабильного пространства

Общие методы оптимизации также позволяют верифицировать нейронную сеть, при этом любая проблема выполнимости преобразуется в задачу оптимизации. Затем становится возможным применить обычные методы оптимизации, такие как алгоритм "ветвления и границы" (Branch and Bound), чтобы решить эту проблему.

Свойство максимально стабильного пространства, как правило, выражается в виде булевой формулы над линейными неравенствами. Например, выход сети должен быть больше некоторой части входного пространства. Чтобы доказать это с помощью метода оптимизации, предполагаемое свойство выражается в виде дополнительных слоев в конце сети. Таким образом, после нахождения решения задачи оптимизации осуществляется решение проблемы выполнимости путем проверки знака решения. В [23] описано построение данной задачи оптимизации на основе задачи выполнимости в нейронной сети, сочетающей классический градиентный спуск для нахождения локального минимума, а также оптимизатор ветвей и границ для определения глобального оптимума.

Еще один пример методов оптимизации для доказательства устойчивости нейронных сетей касается использования программирования с ограничениями [24]. Вначале нейронная сеть аппроксимируется ее моделированием как линейной программы (с использованием сети, составленной из кусочно-линейных функций), затем аппроксимируя возможные состояния, применяя только выпуклые множества и итеративно решатель ограничений, чтобы доказать свойство робастности.

6.3.4 Использование абстрактной интерпретации для доказательства свойства максимального стабильного пространства

Абстрактная интерпретация - это вид формального метода, основанного на теории построения контролируемых аппроксимаций (см. приложение В для получения дополнительной информации). Данный метод часто используют для доказательства сложных свойств программ [25]. Абстрактные интерпретации занимают значительное место в сообществе верификации и валидации программного обеспечения, особенно в контексте критически важного для безопасности программного обеспечения, такого как встроенное программное обеспечение для самолетов [26], автомобилей [27] и космических аппаратов [28].

В работах [29], [30], [31], [32] представлены конструкции новых абстрактных областей, специально адаптированных под поведение нейронных сетей. Нелинейная природа нейронных сетей имеет тенденцию делать некоторые из существующих абстрактных областей неэффективными, особенно это касается областей, использующих аффинную динамику системы для определения абстрактных областей. Так обстоит дело, например, с новой зонотопической областью, описанной в [30], которая отражает специфическую динамику функций активации ReLU, обычно используемых в нейронных сетях обработки изображений.

Для того чтобы доказать устойчивость принимаемого решения по некоторому региону входного пространства, нужно сначала выразить регион входного пространства, подлежащий анализу, с использованием абстрактной области. Затем необходимо определить абстрактную семантику, способную выполнять символьные вычисления нейронной сети в данной абстрактной области. Выходом является абстрактное значение, представляющее аппроксимацию возможных выходных данных классификатора на этом регионе входного пространства. Результатом служит вектор, элементами которого становится достоверность классификации для каждого класса. Любой элемент сам по себе абстрактное значение (например, интервал). После вычисления выхода возможны два случая: либо

один из элементов больше другого, либо отсутствует класс, который бы доминировал при принятии решения.

7 Эмпирические методы

7.1 Общие положения

Следующим аспектом робастности, охватываемым в настоящем стандарте, является субъективная оценка робастности. Данная оценка основана на функциональной оценке на системном уровне, для которой наиболее приемлемы эмпирические методы, с помощью которых проводят сбор сведений по этому вопросу.

7.2 Эксплуатационные испытания

Хотя существует несколько аспектов, которые необходимо изучить при дальнейшем использовании систем ИИ, количество возможных способов анализа поведения и эффективности системы ограничено. Системы ИИ обычно в значительной степени состоят из программного обеспечения, поэтому необходимы стандарты для его тестирования, такие как ISO/IEC/IEEE 29119 [33].

Основные цели тестирования программного обеспечения сформулированы в ISO/IEC/IEEE 29119-3:2013: "Следует предоставить информацию о качестве элемента тестирования и любом остаточном риске в отношении того, насколько элемент тестирования протестирован для обнаружения дефектов в элементе тестирования до его введения в эксплуатацию и для снижения рисков низкого качества продукции для заинтересованных сторон".

Рабочий процесс оценки робастности нейронной сети, изображенный на рисунке 1, состоит из трех следующих шагов, которые имеют решающее значение для каждого эксплуатационного испытания:

1) подготовка плана тестирования (plan testing);
2) сбор данных (data sourcing);
3) проведение испытания в реальных условиях эксплуатации (conduct testing).

В отличие от других методов тестирования, при эксплуатационных испытаниях нейронная сеть интегрируется в систему, которая работает в реалистичной среде для соответствующего приложения. Система также должна реализовывать сбор данных, поэтому поиск и сбор данных являются неотъемлемой частью проектирования и проведения экспериментов.

Дефекты и низкое качество продукции также вызывают беспокойство при тестировании систем ИИ. Однако отказ системы ИИ в функциональном тесте не обязательно связан с ошибкой ("software bug") программного обеспечения или с ошибочным дизайном. При этом системы ИИ, демонстрирующие случайные сбои, иногда используют, поскольку их по-прежнему считают полезными для достижения предполагаемой цели, в частности в тех случаях, когда отсутствуют реальные альтернативы. Системы ИИ эффективны в основном во время эксплуатационных испытаний или при внедрении, например в случае с такими системами, как виртуальные помощники, что относится ко многим системам ИИ, функционирующим во взаимодействии с природной средой и пользователями или зависящим от них.

Вопросы разрешения неопределенности в отношении эффективности продукта и рисков, связанных с его внедрением, - предмет многих нормативных актов в области медицины. Например, в Европе медицинские устройства, в том числе с использованием ИИ, должны соответствовать ИСО 14155. Порядок прохождения клинической оценки или клинических испытаний программного обеспечения с применением ИИ, являющегося медицинским изделием, определяется национальным или региональным законодательством [34], [35], [36].

Для немедицинских устройств, использующих ИИ, эксплуатационные испытания в течение продолжительного времени являются признанным средством сравнения и оценки робастности решений. Вот несколько примеров:

- испытания на распознавание лиц [37], [38], [39];
- тестирование систем поддержки принятия решений для сельскохозяйственных приложений [40];
- практика испытаний беспилотных автомобилей [41];
- тестирование систем распознавания речи и голоса [42], [43];
- сетевой робот на вокзале [44].

Эксплуатационные испытания систем ИИ различаются по методологии, количеству пользователей или использованных образцов, статусу ответственной организации/лиц и документации результатов.

7.3 Апостериорное тестирование

В некоторых случаях можно формально подтвердить робастность интеллектуальной системы. Когда это невозможно, что часто бывает с нейронными сетями [45], выполняют валидацию путем эмпирического тестирования робастности системы, и оценка на основе ввода/вывода востребована в данном контексте. В таком виде оценки существуют методы априорного тестирования и апостериорного тестирования. В то время как при априорном тестировании ожидаемый результат известен, и поэтому применимы статистические показатели, при апостериорном тестировании результат заранее неизвестен. В этом случае возможно предпринять автоматизированные действия, чтобы по-прежнему проводить статистические измерения косвенными средствами. В противном случае единственным доступным методом является эмпирический, основанный на суждении людей.

При апостериорном тестировании шаги 4 и 5 процесса, изображенного на рисунке 1, слегка изменены. Шаг 4, вероятно, будет более сложным, потому что правильный ответ заранее неизвестен. Интерпретация результатов на шаге 5 - это, скорее всего, предмет консенсуса, а не однозначной истины.

Как правило, для проверки робастности системы определяют данные или тестовые среды, представляющие широкий спектр тестовых сценариев для нормальных условий эксплуатации и критических случаев (шаг 2 процесса). Эти входные данные передаются в систему для оценки, а выходные данные системы (называемые гипотезами) сравниваются с эталонами, то есть с достоверной информацией (шаг 3). Входные данные предназначены для того, чтобы внести возмущение в систему для проверки ее робастности, например, используя неблагоприятные примеры. Такие эталоны обычно предоставляются экспертами, выполняющими такую же задачу, как и оцениваемая система, или являются результатом физических измерений.

В случае априорного тестирования эталоны ссылки предоставляются экспертами, выполняющими аннотации, и обычно они договариваются друг с другом в отношении правильного ответа, который должен быть получен (высокая степень согласия между экспертами). В таком случае эталон (ground truth) определяется однозначно. Напротив, при апостериорном тестировании эталоны, создаваемые экспертами, варьируются, поэтому эталон эксплуатационных испытаний неоднозначен, так как у задачи есть несколько правильных ответов [46].

Поскольку невозможно определить решение априори все возможные правильные ответы, поэтому выполняют апостериорные оценки. То есть при рассмотрении входных данных систем эксперты, предоставляющие аннотации (автоматизированные измерители), могут установить, являются ли они правильными или неправильными.

Машинный перевод - классический пример той задачи, для которой апостериорная оценка служит полезным дополнением к априорному тестированию. Обычно существуют различные способы перевода одного и того же предложения с одного языка на другой. Хотя в данном случае часто применяют статистические методы путем установления произвольного набора правильных или приемлемых ответов для сравнения результатов [47], это не является полностью надежным показателем эффективности, и субъективное апостериорное тестирование часто бывает более точным. Также применительно к навигационной задаче можно использовать несколько траекторий для перемещения из одного места в другое. В зависимости от способности определить объективный критерий оптимальных траекторий, апостериорное тестирование может быть выполнено либо статистическими, либо эмпирическими средствами.

Также возможно использовать апостериорную оценку для валидации новой робастной метрики (новый метод или формула для измерения). Когда качество задачи является субъективным, метрикам необходимо присвоить баллы качества, которые коррелируют с пользовательским мнением о качестве. Суждение пользователей - это эталон для оценки автоматических метрик [48].

Однако концепции апостериорной оценки и оценки после развертывания системы пересекаются в некоторых случаях, особенно при тестировании с конечными пользователями. Например, в случае оценки качества взаимодействия человека с машиной оценку выполняют апостериорно, поскольку невозможно установить, каким образом это взаимодействие будет оказывать влияние на все слои населения до того, как оно получит широкое распространение. Для проведения такой оценки можно варьировать профиль пользователя, иметь пул пользователей, адекватно отражающий фактические условия работы системы, и получать с его помощью эмпирический анализ робастности этой интерактивной интеллектуальной системы.

7.4 Эталонное тестирование нейронных сетей

Эталонное тестирование (бенчмаркинг, benchmarking) системы, основанной на нейронных сетях, может способствовать определению степени робастности

системы. Часто первоначальное доверие к решению ИИ, основанному на нейронных сетях, устанавливается с помощью эталонного тестирования. Например, продолжительное время в распознавании образов и аналогичных применениях методов ИИ эталонное тестирование было наиболее оптимальным решением для установления доверия к определенному методу [49]. Вместе с тем, проведение эталонного тестирования может иметь элементы субъективности, например при маркировке или аннотировании тестовых наборов данных экспертами-практиками.

Эталонное тестирование измеряет производительность системы на основе тщательно разработанных наборов данных, которые в большинстве случаев являются общедоступными. Часто их используют для тестирования различных систем. Наиболее приемлемыми примерами эталонного тестирования являются тесты поставщиков по распознаванию лиц (face recognition vendor tests, FRVT), проведенные Министерством торговли США [50]. Другие примеры приведены в работе "Большие вызовы в биомедицинском анализе изображений" [51].

В отличие от 7.2, эталонное тестирование необязательно требует наличия действующей системы в реальных условиях применения. Для целей сопоставления создают такие наборы данных, использование которых вызывает существенные вопросы при применении современных методов классификации или регрессии. Наборы контрольных данных должны быть дополнены набором правил эталонного тестирования, которые описывают и стандартизируют способы настройки тестирования, документирования этих настроек, измерения и документирования результатов [52].

Эталонное тестирование имеет существенное значение при проведении исследований в области распознавания образов и вносит решающий вклад в развитие этой области. Однако эталонного тестирования обычно недостаточно для определения целей робастности. Результаты сравнительного анализа следует интерпретировать с предельной внимательностью [53].

Приложение А
(справочное)

Возмущение данных

А.1 Общие положения

Возмущение данных (data perturbation) формально определяют как гомоморфизм (т.е. отображение из заданной области в себя) над областью возможных входов системы. Примером такой области является область входных данных, содержащая все изображения RGB определенной ширины и высоты. В этом приложении описаны возмущения набора данных в контексте оценки устойчивости нейронных сетей.

Например, в автоматизированных системах классификации широко использованы промышленные нейронные сети. Такие системы классификации применяют для распознавания лиц, отслеживания объектов, распознавания звука и т.д. Обычным способом построения системы классификации на основе нейронных сетей является выполнение контролируемого обучения с помощью размеченной базы данных.

Даже последние версии нейронных сетей для решения задачи классификации весьма восприимчивы к искажениям данных или к неблагоприятным примерам [54]. Неблагоприятные (состязательные, adversarial) примеры включают изображения или звуковые образцы, которые немного изменены по сравнению с оригиналами, что приводит к другому результату классификации, и возникают естественным образом в окружающей среде или из-за свойств датчиков. Существует множество методов построения таких примеров, но в настоящее время отсутствует приемлемый способ их обнаружения.

С точки зрения инженеров-программистов ИИ, существование состязательных примеров представляет риск для робастности системы, поскольку в некоторых случаях система ведет себя нестабильно. Инженеры знают о наличии состязательных примеров, однако их нелегко выявить заранее.

Применение состязательного примера для провокации незапланированного поведения нейронных сетей может представлять собой атаку. В литературе встречаются две основные парадигмы таких атак:

- атака "белого ящика" (white-box attack), при которой злоумышленник имеет полное знание нейронной сети, обучающий набор данных и алгоритм обучения;
- атака "черного ящика" (black-box attack), при которой злоумышленник не знает архитектуры нейронной сети, набора обучающих данных или алгоритма обучения.

Хотя в этом приложении описаны различные типы возмущений для разных типов данных, оно не претендует на то, чтобы быть исчерпывающим. Также следует отметить, что аппаратные средства могут вызывать незапланированное поведение, изменяя данные в ходе числовых преобразований и, таким образом, приводя к искажениям. В литературе также предложены стратегии и методы защиты систем от этих типов атак.

В А.2 и А.3 представлены примеры искажений данных для изображений и звуков. В каждом случае как случайные естественные возмущения, так и преднамеренные атаки сосуществуют в широком диапазоне применений.

А.2 Примеры искажений изображений

А.2.1 Общие положения

Существует несколько типов возмущений изображения, которые могут отражать возможную деградацию, которую окружающая среда способна нанести изображению, обрабатываемому системой ИИ. Изображение (обычно) представляет собой двумерный массив из пикселей, каждый из которых представлен одним или несколькими числовыми значениями (например, одно - для черного изображения и пикселей, три - для изображения RGB). Без потери общности ниже будет рассмотрено изображение как массив пикселей шириной *W* и высотой *H*, в то время как каждый пиксель $\rho_{i,j}$ находится между значениями от 0 до 255. Следовательно, изображение является точкой в пространстве размером $L \times W$. Возмущение изображения - это функция, которая преобразует одно изображение в другое.

Когда два изображения находятся в одном пространстве, доступны различные метрики для расчета расстояния между ними, включая среднеквадратичную ошибку, расстояние Левенштейна [55], индекс структурного сходства [56] и т.д. Каждое возмущение применимо также к тем цветным изображениям, в которых возмущение применяется к каждому цветовому каналу. Существует как множество возможных отклонений, так и множество метрик, которые определяют, какие из результатов оказываются ближе к исходному изображению. Атаки могут быть разработаны для имитации фактического ухудшения процесса получения изображения, например шума, вибрации, ослепления или преграждения объектива камеры.

В А.2.2-А.2.7 приведены некоторые примеры возмущений изображения и некоторые метрики, используемые для их оценки.

А.2.2 Однородный шум

Однородный шум - это преобразование, которое добавляет ограниченное случайное возмущение каждому пикселю изображения. Однородный шум определяется значением *K*, соответствующим максимуму шума, который можно применить к каждому пикселю. Операцию сложения или вычитания значения шума выбирают случайным образом для каждого пикселя.

Формально каждый исходный пиксель $p_{i,j}$ преобразуется с помощью следующей функции:

$$p_{i,j} = v \rightarrow p'_{i,j} = v \pm k, \quad \text{(А.1)}$$

где $p_{i,j}$ - исходный пиксель;

v - переменная, в которую записывается яркость текущего пикселя;

$p'_{i,j}$ - преобразованный пиксель;

k - случайная величина шума, такая, что $k \leq K$, $0 \leq v \pm k \leq 255$;

K - максимальный шум, который можно применить к каждому пикселю.

В случае однородного шума используемая метрика напрямую коррелирует со значением *K*. Изображение l_1 находится на расстоянии 1 от изображения l_2, если:

- для каждого пикселя $p^1_{i,j}$ из l_1 и соответствующего ему пикселя $p^2_{i,j}$ из l_2, $| p^1_{i,j} - p^2_{i,j} | \leq k$ и

- существует хотя бы один пиксель (*i*, *j*), для которого $| p^1_{i,j} - p^2_{i,j} | = k$.

А.2.3 Осветление (brightening)

Осветление (или его обратное преобразование - затемнение) - одно из наиболее простых доступных преобразований изображения. Оно используется для изменения освещенности изображения. Условия освещенности могут влиять на эффективность классификатора изображений [57]. По этой причине крайне важно продемонстрировать устойчивость классификатора к такому возмущению.

Формально каждый пиксель $p_{i,j}$ преобразуется, например, с помощью следующей функции:

$$p_{i,j} = v \rightarrow p'_{i,j} = v + k, \quad \text{(A.2)}$$

где $p_{i,j}$ - исходный пиксель;

v - переменная, в которую записывается яркость текущего пикселя;

$p'_{i,j}$ - преобразованный пиксель;

k - случайная величина, такая, что $-255 \leq k \leq 255$.

Другой способ настроить параметр яркости для изображения RGB - переключиться на представление HSL (оттенок/насыщенность/яркость) и изменить представление канала яркости.

В случае искажения осветления соответствующей метрикой является либо канал яркости представления HSL, либо константа *k*, используемая для применения возмущения к представлению в оттенках серого или RGB.

А.2.4 Вибрация и вращение

Когда камеры установлены на крыше транспортного средства, происходит ухудшение изображения из-за вибрации или колебаний устройства. Эти возмущения вызывают некоторую инверсию (inversion) пикселя от одного кадра к другому.

При рассмотрении устойчивости к таким возмущениям вращения или вибрации необходимо рассмотреть набор преобразованных такими возмущениями изображений. Для поворота используют метрику максимального угла поворота, применяемого к изображению. Модель повернутого изображения [30] реализована путем первоначального вычисления действительного положения *i*', *j*', которое должно быть отображено в центр выбранного пикселя. Затем вычисляют яркость повернутого пикселя путем выполнения линейной интерполяции с учетом соседних пикселей, так что вклад каждого пикселя пропорционален расстоянию до *i*', *j*', отсекая вклад на расстояние 1.

Можно смоделировать вибрацию изображения, используя адекватное ядро размытия (blurring kernel) [58], которое представляет степень вибрации, влияющей на изображение. Эта модель позволяет предотвратить дрожание изображения путем устранения размытия изображения после обнаружения соответствующего ядра размытия. Для вибрации и мерцания можно использовать метрику [59], построенную с помощью процесса Кокса. Процесс Кокса служит моделью для агрегирования паттернов пространственных точек, когда эти паттерны следуют стохастическому процессу.

А.2.5 Атмосферная турбулентность

Оптические искажения и шумы иногда возникают из-за влияния турбулентного потока воздуха, изменений температуры, плотности частиц в воздухе, влажности и уровня диоксида углерода. С учетом этих факторов преломление света изменяется при пересечении нескольких слоев воздуха с различными свойствами. В результате получается геометрическое искажение, которое значительно ухудшает способность обработки изображения, что характерно в том случае, когда полученное изображение проходит через большое количество атмосферных слоев, например: в случае спутникового изображения или изображения с большой высоты - с помощью беспилотного летательного аппарата. В частности, в [60] и [61] описана структура модели атмосферной турбулентности. Связанная с этим проблема заключается в улучшении восстановления изображений для удаления атмосферной турбулентности [62], [63], [64]. Для этих методов следует использовать метрики, моделирующие интенсивность возмущения.

А.2.6 Размытие

В компьютерном зрении размытие (blurring) используют для сглаживания краев изображения. Наиболее распространенный эффект размытия достигается с помощью размытия по Гауссу, но можно определить несколько других эффектов размытия, применяя следующий принцип. Размытие по Гауссу, как правило, основано на свертке изображения с ядром из определенных значений. Размер ядра выбирают таким образом, чтобы достичь большего или меньшего эффекта размытия.

Коэффициенты ядра определяют стандартным отклонением ядра или вручную для применения негауссова эффекта [например, блочного размытия (Box blur)].

Формально размытие определено ядром *K* размером $n \times m$, обычно $n=m=3$. Каждое возмущенное изображение строится путем применения ядра к каждому пикселю исходного изображения и сохранения результата свертки как значения нового пикселя. В [65] авторы представляют некоторые общие эффекты размытия

(движение, расфокусировку и гауссов шум), а также способ их автоматической классификации.

В случае размытия по Гауссу соответствующими метриками являются размер ядра и стандартное отклонение ядра. Другие эффекты размытия приводят к другим метрикам в зависимости от их параметризации. Каждая метрика позволяет задать интенсивность размытия, применяемого к изображению.

А.2.7 Блуминг

Блуминг (blooming) определяют как образование яркого пятна, которое распространяется вокруг локальной передержки (переэкспозиции). Причина артефакта блуминга заключается в том, что отдельные светочувствительные элементы ПЗС- или КМОП-сенсора (пикселя) иногда могут поглощать только ограниченное количество света. Если величина заряда такого пикселя превышает заданный порог вследствие точечной передержки, его избыточный заряд передается соседним пикселям. Заряды поступают в основном на пиксели, которые соединены друг с другом для переноса заряда при считывании с датчика. Поскольку эти пиксели чаще всего связаны в вертикальном направлении, артефакт часто возникает в виде четко определенной полосы, которая кажется полностью переэкспонированной [66], [67].

Формально блуминг определен ядром K размером $n \times m$ вокруг локального пятна, где $0 \leq n \leq L$ и $0 \leq m \leq W$ и в зависимости от направления связанных пикселей $m>>n$ или $n<<m$. Каждый пиксель $p_{i,j}$ в ядре K преобразуется с помощью следующей функции:

$$p_{i,j} = v \rightarrow p'_{i,j} = v = 255, \quad \text{(A.3)}$$

где $p_{i,j}$ - исходный пиксель;

v - переменная, в которую записывается яркость текущего пикселя;

$p'_{i,j}$ - преобразованный пиксель.

Датчики со специальными структурами, препятствующими засветке, могут обойти этот артефакт, но имеют недостаток, заключающийся в том, что из-за дополнительного необходимого места для этих структур уменьшается размер пикселя, а вместе с ним - и чувствительность датчика [68].

А.2.8 Размытие (Smear)

Размытие определяют как образование ярких линий на изображении, которые являются вертикальными для медленных камер CCD (charge-coupled device) или имеют угол к вертикали для камер CCD с более высокой скоростью. Причиной артефакта размытия является перенос заряда после экспонирования, который приводит к экспонированию передающих сигнал пикселей в случае попадания яркого пучка света [69], [70]. В отличие от наиболее резко очерченной полосы при блуминге, полоса с эффектом размытия всегда достигает края изображения и не кажется полностью переэкспонированной.

Формально вертикальное пятно определено полосой S размером $L \times m$ с равномерной интенсивностью вдоль каждого столбца, начиная с передержанного пятна. Каждый пиксель $p_{i,j}$ преобразуется в S с помощью следующей функции:

$$p_{i,j} = v \rightarrow p'_{i,j} = v + b, \quad \text{(A.4)}$$

где $p_{i,j}$ - исходный пиксель;

v - переменная, в которую записывается яркость текущего пикселя;

$p'_{i,j}$ - преобразованный пиксель;

b - величина, на которую увеличивается яркость в S, $0 \leq v+b<255$.

Устранение размытия до появления тех артефактов, которые возникают перед новой экспозицией, осуществляется путем сброса заряда. Но артефакты, образующиеся после передержки, можно обойти только с помощью механического или электрооптического затвора [70].

А.3 Примеры звуковых возмущений

А.3.1 Основы

Звуковые возмущения могут влиять на широкий спектр систем обработки звука. Значительное количество систем основано на вводе голосовых команд как первичном интерфейсе управления устройством. Например, виртуальные помощники с функциями автоматического распознавания речи (Automatic Speech Recognition, ASR), устанавливаемые на домашних или мобильных устройствах, способны управлять освещением, домашней утварью, системами безопасности дома, а также осуществлять покупки и помогать с телефонными звонками и отправкой сообщений. Однако такие системы могут быть чувствительны к возмущениям входных звуковых данных, что потенциально может привести к неспособности распознать команду, незапланированному или даже к злоумышленному действию, если возмущение связано с атакой.

Имеется два основных типа звуковых возмущений:

- модифицирующих акустический сигнал в частотном диапазоне слышимости человеком;
- основанных на ультразвуке.

В обзорных работах [71], [72] изложены методики для этих типов возмущений.

A.3.2 Звуковые возмущения в частотном диапазоне слышимости человеком

Одними из первых работ, в которых рассматривались автоматизированные атаки на входной речевой сигнал, являются работы [73] и [74]. Возмущенные сигналы воспринимаются как определенные команды для системы автоматического распознавания речи, но не воспринимаемы человеком, и в некоторых случаях даже не заметны человеку. Эти методы, по сути, изменяют акустические характеристики сигнала, на которые ориентируются многие системы распознавания речи (например, Mel-frequency cepstral coefficients).

В [75] описано возмущение сигнала, основанное на генетическом алгоритме, которое применяется против облегченной/простой ASR. В [76] представлена усовершенствованная версия этого возмущения, основанная на современной системе DeepSpeech. Метод SirenAttack, изложенный в [77], является широко применимой атакой на основе оптимизации роем частиц.

В [78] приведен пример речевой атаки, в котором незначительные возмущения исходной речи приводят к ошибочной работе Mozilla DeepSpeech ASR. Возмущения заданы математической оптимизационной методикой, формирующей сигнал, воспринимаемый как иной заранее заданный текст. Это атака типа "белый ящик", т.е. методика требует знаний об атакуемой системе.

В [79] разработаны неощутимые атакующие аудиосигналы (что проверено в ходе эксперимента с людьми) путем задействования психоакустических принципов слуховой маскировки. Такие сигналы обладают 100%-ной эффективностью для произвольных полнофразовых целей (команд). В этой работе также достигнут прогресс в создании примеров аудиопротиводействия в физическом мире путем построения возмущений, которые остаются эффективными даже после применения реалистично смоделированных искажений окружающей среды. В [80] предложена аналогичная идея с использованием психоакустического скрытия для атаки на систему ASR. Оба эти метода являются атаками "белого ящика".

В [81] демонстрируется существование универсальных атакующих аудиовозмущений, которые приводят к ошибочной транскрипции аудиосигналов средствами систем распознавания речи. Предложен алгоритм нахождения единичного квазинеощутимого возмущения, которое при добавлении к произвольному речевому сигналу, скорее всего, может привести к ложному срабатыванию модели распознавания речи. Этот метод применен к системе распознавания речи Mozilla DeepSpeech. Также в работе показано, что подобные возмущения обобщаются на значительное количество других моделей, не доступных в ходе обучения. Например, с помощью теста на переносимость показана применимость атаки на ASR, основанные на WaveNetç .

Подход, описанный в [82], показывает возможность отдачи скрытых голосовых команд путем их внедрения в песни, проигрывание которых позволяет в существенной мере управлять целевой системой через распознавание речи без обнаружения.

A.3.3 Атаки, основанные на ультразвуке

Ультразвуковые атаки в основном опираются на нелинейность записывающего устройства, приводя к записи неслышимого звука. Впервые этот эффект был достигнут в работе [83], в которой модулированные ультразвуковые передачи преобразованы через микрофон и нелинейный усилитель в корректные команды, исполняемые в разнообразных коммерческих системах распознавания речи. В работе [84] также отмечено, что нелинейности в динамиках усложняют атакующему задачу по увеличению радиуса атаки, поэтому использовано множество динамиков в виде массива ультразвуковых динамиков (ultrasonic loudspeaker array) для проведения атаки на ASR на большом расстоянии.

В [85] применены неслышимые ультразвуковые передачи для записи слышимых сигналов с помощью скрытого канала с высокой пропускной способностью (high-bandwidth covert channel) в своем методе BackDoor. В работе [86] развиты идеи метода BackDoor, что не требует программного обеспечения для микрофона,

поэтому возмущение можно использовать на различных развернутых микрофонах или ассистентах.

Кроме того, в работе [87] рассмотрены атаки на системы классификации звуков, в настоящее время использующих только ультразвуковой диапазон.

Приложение В
(справочное)

Принцип абстрактной интерпретации

B.1 Принцип абстрактной интерпретации

Принцип представлен следующим образом. Во-первых, этот подход обычно рассматривает все возможные варианты (следы) выполнения программы с использованием семантики, например детонационная [88] или аксиоматическая [89] семантика. Набор всех следов выполнения, выраженный семантикой, образует решетку (полный частичный порядок, определенный для набора всех следов) или, по крайней мере, набор частичного порядка. Такая решетка называется конкретной областью и, как известно, является трудноразрешимой. Затем определяется вторая область, называемая абстрактной областью, поскольку это абстракция конкретной области. Абстрактная область также представляет собой решетку или, по крайней мере, набор частичного порядка. Оказывается, что абстракция верна, когда связь Галуа определена (и доказана) между двумя областями. Связь Галуа осуществляется путем определения двух конкретных функций: одна называется абстракцией α (переход от конкретной области к абстрактной), а другая - конкретизацией γ (переход от абстрактной области к конкретной). Эти функции обладают специфическими свойствами, которые необходимо доказать заранее, в основном монотонность α и γ, расширяемость $\gamma^{\circ}\alpha$ и сжимаемость $\alpha^{\circ}\gamma$.

Как только связь Галуа между двумя областями доказана, абстрактная область становится осмысленным чрезмерным приближением всех конкретных исполнений, которое также является разрешимым (tractable, по построению). Затем можно доказывать свойства на абстрактной области и автоматически переносить их на соответствующие конкретные следы, представленные абстракцией. Основная трудность при абстрактной интерпретации состоит в том, чтобы определить достаточно простую, но выразительную абстрактную область. Абстрактная область предназначена быть как разрешимой, так и репрезентативной по отношению к конкретным следам системы. Существует обширный объем публикаций по определениям абстрактных областей для представления численных расчетов. Например, можно использовать интервалы [90], пятиугольники [91], восьмиугольники [92], шаблоны [93], многогранники [94], зонотопы [95] и т.д. Каждый из вариантов является результатом различного компромисса между точностью абстракции и стоимостью ее расчета.

Пример. Рассмотрим черно-белое изображение размером 2x2, каждый пиксель (p_1, p_2, p_3, p_4) находится в диапазоне от 0 до 255. Предположим, что существует обученная нейронная сеть, которая выполняет классификацию таких изображений между классами A и B, а также то, что изображение *I* со значениями (100, 100, 50, 150) классифицируется как A с достоверностью 90% и B с достоверностью 10%. Наконец, предположим, что к изображению применяется однородный шум с интенсивностью 5. Целью обеспечения стабильности нейросети является доказательство того, что любое изображение *I*', полученное из *I* посредством добавления такого однородного шума, также в первую очередь относится к классу A. Используя интервальную абстрактную область и определение однородного шума, строят следующее абстрактное изображение $I^{\#}$ со значениями ([95; 105], [95; 105], [45; 55], [145; 155]). Изображение $I^{\#}$ представляет все возможные изображения *I*', которые можно получить из *I* с однородным шумом с интенсивностью, равной 5. Например, оно представляет изображения (102, 104, 45, 150), (98, 100, 53, 155)... Количество изображений, представленных $I^{\#}$, достаточно велико, в данном случае это $(5\times 2+1)^{2\times 2}$, в общем случае это $(K\times 2+1)^{L\times W}$ (с максимальной интенсивностью *K* примененного однородного шума, $L\times W$ - это размер изображений). Используя абстрактную семантику, можно вычислить абстрактный вывод нейронной сети. В этом случае предположим, что результат будет следующим ([75; 92], [7; 34]). Первая часть - это достоверность классификации класса A для изображения $I^{\#}$, а вторая часть - для класса B. Поскольку изображение *I* также представлено в $I^{\#}$, значение 90 принадлежит отрезку [75; 92], а значение 10 также содержится в отрезке [7; 34]. В этом случае все изображения, представленные $I^{\#}$, всегда имеют строго большую уверенность в классе A, чем в классе B, поэтому классификация сети не меняется ни на одном из этих изображений. Однако минимум достоверности класса A составляет 75, что в некоторых случаях ниже порога,

необходимого для присвоения класса. Это допускает, что определенные изображения, представленные $I^{\#}$, не отнесены к какому-либо классу. Это не обязательно, поскольку абстрактное значение является переаппроксимацией конкретного результата. Чтобы иметь более жесткие границы возможных выходных данных, требуется более сложные абстрактные области, например зонотопическая область вместо интервальной области.

Библиография

[1] ISO 26262:2018 Road vehicles Functional safety

[2] Aeronautics, The Radio Technical Commission for Software Considerations in Airborne Systems and Equipment Certification. The Radio Technical Commission for Aeronautics. 2012

[3] ISO/IEC/IEEE 16085:2020 Systems and software engineering Life cycle processes Risk management

[4] ISO 14155:2011 Clinical investigation of medical devices for human subjects Good clinical practice

[5] Goodfellow I.J., Shlens J., Szegedy C. Explaining and harnessing adversarial examples. Astrophysics data system. 2014, arXiv: 1412.6572

[6] Liang D., Hayes P., Althof A. Deep Adversarial Robustness. 2017

[7] Yuan X., He P., Zhu Q., Li X. Adversarial Examples: Attacks and Defenses for Deep Learning. IEEE Transactions on Neural Networks and Learning Systems. 2019

[8] Fawcett T. An Introduction to ROC Analysis. Elsevier Science Inc., Pattern Recognition Letters, Vol. 27. 2006

[9] David M.W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, Vol. 2. 2011

[10] Ting K.M. Encyclopedia of machine learning. Springer. 2011, ISBN 978-0-387-30164-8

[11] Brooks H., Brown B., Ebert B., Ferro C., Jolliffe I., Koh T.Y., Roebber P., Stephenson D. WWRP/WGNE Joint Working Group on Forecast Verification Research. World Meteorological Organisation, Collaboration for Australian Weather and Climate Research. 2017

[12] Brodersen K.H., Ong C.S., Stephan K.E., Buhmann J.M. The Balanced Accuracy and Its Posterior Distribution. Istanbul: IEEE, 20th International Conference on Pattern Recognition. 2010, ISBN 978-1-4244-7541-4

[13] Tsoumakas G., Katakis I., Vlahavas I. Data Mining and Knowledge Discovery Handbook, in Mining Multi-label Data. Springer, Boston, MA. 2009, ISBN 978-0-387-09822-7

[14] Matthews B.W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure, Vol. 405. 1975, ISSN 0005-2795

[15] Chicco D. Ten quick tips for machine learning in computational biology. BioData Mining, Vol. 10. 2017, PMCID: PMC5721660

[16] Crammer K., Singer Y. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. JMLR, Journal of Machine Learning Research, Vol. 2. 2001, ISSN 1532-4435

[17] Choi J.Y., Choi C.H. Sensitivity analysis of multilayer perceptron with differentiable activation functions.1, IEEE, Transactions on Neural Networks, Vol. 3. 1992, ISSN 1045-9227

[18] Montaño J.J., Palmer A. Numeric sensitivity analysis applied to feedforward neural networks. Springer, Neural Computing & Applications, Vol. 12. 2003, ISSN 1433-3058

[19] Hess D.E., Roddy R.F., Faller W.E. Uncertainty Analysis Applied to Feedforward Neural Networks. Applied Simulation Technologies, Vol. 54. 2007

[20] Katz G., Barrett C., Dill D., Julian K., Kochenderfer M. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. Springer, Computer Aided Verification. 2017

[21] Huang X., Kwiatkowska M., Wang S., Wu M. Safety Verification of Deep Neural Networks. Springer, Computer Aider Vision. 2016

[22] Ehlers R. Formal verification of piece-wise linear feed-forward neural networks. Automated Technology for Verification and Analysis. 2017

[23] Bunel R., Turkaslan I., Torr P., HS., KOHLI P., KUMAR M.P Piecewise linear neural network verification. 2017, CoRR

[24] Bastani O., loannou Y., Lampropoulos L., Vytiniotis D., Nori A., Criminisi A. Measuring Neural Net Robustness with Constraints. Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016, ISBN 978-1-5108-3881-9

[25] Cousot P., Cousot R. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. ACM, Conference Record of the Fourth

Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. 1977

[26] Souyris J., Delmas D. Experimental Assessment of Astrée on Safety-Critical Avionics Software. Springer, Proceeding of International Conference on Computer Safety, Reliability, and Security, Vol. 4680. 2007

[27] Yamaguchi T., Brain M., Ryder C., Imai Y., Kawamura Y. Application of Abstract Interpretation to the Automotive Electronic Control System. Springer, Verification, Model Checking, and Abstract Interpretation, Vol. 11388. 2019

[28] Bouissou O., Conquet E., Cousot P., Cousot R., Feret J., Ghorbal K., Goubault E., Lesens D., Mauborgne L, Miné A., Putot S., Rival X., Turin M. Space software validation using Abstract Interpretation. Data Systems in Aerospace, Vol. 669. 2009

[29] Gehr T., Mirman M., Drachsler-Cohen D., Tsankov P., Chaudhuri S., Vechev M.T. AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation. IEEE Symposium on Security and Privacy, Vol. 2018, ISSN 2375-1207

[30] Singh G., Gehr T., Püschel M., Vechev M. An Abstract Domain for Certifying Neural Networks. ACM, Proceedings of the ACM on Programming Languages. 2019, ISSN 2475-1421

[31] Mirman M., Gehr T., Vechev M. Differentiable Abstract Interpretation for Provably Robust Neural Networks. Proceedings of the 35th International Conference on Machine Learning. 2018

[32] Pulina L., Tacchella A. An Abstraction-Refinement Approach to Verification of Artificial Neural Networks. Springer, Computer Aided Verification. Vol. 6174. 2010, ISBN 978-3-642-14295-6

[33] ISO/IEC/IEEE 29119-3:2013 Software and systems engineering Software testing Part 3: Test documentation

[34] Proposal for guidelines regarding classification of software based information systems used in health care. The Medical Products Agency's Working Group on Medical Information Systems, Lakemedelsverket Medical Products Agency. 2009, доступно по https://lakemedelsverket.se/upload/foretag/medicinteknik/en/Medical-Information-Systems-Report_2009-06-18.pdf

[35] Florek H.J., Brunkwall J., Orend K.H., Handley I., Pribble J., Dieck R. Results from a First in-Human Trial of a Novel Vascular Sealant. Frontiers in Surgery, Vol. 2. 2015

[36] Beede E., Elliott E., Hersch F., lurchenko A., Wilcox L., Ruamviboonsuk P. et al. A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy. ACM Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. 2020

[37] An investigation into the performance of facial recognition systems relative to their planned use in photo identification documents - BioP. BSI. 2004, доступно из: https://www.bsi.bund.de/SharedDocs/Downloads/EN/SI/Publications/Studies/BioP/BioPfinalreport_pdf.pdf

[38] Automated Border Control (ABC) Trial Stansted Airport BAA British Airports Authority and Accenture. 2009, доступно по https://www.accenture.com/t20150523T054056Z__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Technology_2/Accenture-ABC-Trial-Summary-Report.pdfla=en

[39] Vetter V., Zielke T., von Seelen W. Integrating face recognition into security systems. In: Audio- and Video-based Biometric Person Authentication. Springer Berlin Heidelberg, Vols. Audio- and Video-based Biometric Person Authentication (AVBPA). 1997, доступно из: https://doi.org/10.1007/bfb0016025

[40] Burke J., Dunne B. Field testing of six decision support systems for scheduling fungicide applications to control Mycosphaerella graminicola on winter wheat crops in Ireland. The Journal of Agricultural Science, Vol. 146. 2008

[41] The Pathway to Driverless Cars, A Code of Practice for testing. UK Government, Department for Transportation. 2015, доступно из: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/446316/pathway-driverless-cars.pdf

[42] Lamel L., Gauvain J.L., Bennacef S.K., Devillers L., Foukia S., Gangolf J.J. et al. Field trials of a telephone service for rail travel information. Third IEEE Workshop on Interactive Voice Technology for Telecommunications Applications. 1996

[43] Isobe T., Morishima M., Yoshitani F., Koizumi N., Murakami K. Voice-activated home banking system and its field trial. IEEE, Proceedings Fourth International Conference on Spoken Language, Vol. 3. 1996

[44] Shiomi M., Sakamoto D., Kanda T., Ishi C.T., Ishiguro H., Hagita N. Field Trial of a Networked Robot at a Train Station. Springer, International Journal of Social Robotics, Vol. 3. 2011

[45] Tian Y., Pei K., Jana S., Ray B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. ACM, Proceedings of the 40th international conference on software engineering. 2018, ISBN 978-1-4503-5638-1

[46] Van Slype G. Critical study of methods for evaluating the quality of machine translation. Commission of European Communities Directorate General Scientific and Technical Information and Information Management. 1979, BR 19142

[47] Papineni K., Roukos S., Ward T., Zhu W.J. BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th annual meeting on association for computational linguistics. 2002

[48] Graham Y., Baldwin T. Testing for significance of increased correlation with human judgment. Association for Computational Linguistics, Conference on Empirical Methods in Natural Language Processing. 2014

[49] Kohonen T., Barna G., Chrisley R.L. Statistical pattern recognition with neural networks: benchmarking studies. IEEE, Vol. Proceedings of International Conference on Neural Networks (ICNN'88). 1988

[50] Ngan M., Grother P.J. Face recognition vendor test (FRVT) performance of automated gender classification algorithms. US Department of Commerce, National Institute of Standards and Technology. 2015

[51] Van Ginneken B., Kerkstra S., Meakin J. доступно по https://grand-challenge.org

[52] Prechelt L. PROBEN1: A set of benchmarks and benchmarking rules for neural network training algorithms. Fakultaet fuer Informatik, Universitaet Karlsruhe. 1994

[53] Maier-Hein L., Eisenmann M., Reinke A., Onogur S., Stankovic M., Scholz P., Arbel T., Bogunovic H., Bradley A.P., Carass A. et al. Why rankings of biomedical image analysis competitions should be interpreted with care. Nature Communications, Vol. 9. 2018, Doi: 10.1038/s41467-018-07619-7

[54] Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow I.J., Fergus R. Intriguing properties of neural networks. IEEE, Computer Vision and Pattern Recognition. 2013

[55] Levenstein V.I. Binary Codes with Correction for Deletions and Insertions of the Symbol 1. Problemy Peredachi Informatsii.1965

[56] Wang Z. et al. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, Vol. 13. 2002

[57] Kexin P., Yinzhi C., Junfeng Y., Suman J. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. ACM, Proceedings of the 26th Symposium on Operating Systems Principles. 2017, ISBN 978-1-4503-5085-3

[58] Fergus R., Singh B., Hertzmann A., Roweis S.T., Freeman T.W. Removing Camera Shake from a Single Photograph. ACM, 2006, ACM SIGGRAPH 2006 Papers. 2006, ISBN 1-59593-364-6

[59] Sur F., Grediac M. Measuring the noise of imaging sensors in the presence of vibrations and illumination flickering: modeling, algorithm, and experiments. INRIA, MAGRIT, Institut Pascal. 2015, hal-01104124, RR-8672

[60] Li Y., Iwamoto Y., Ogawa K., Chen Y.-W. Computer Simulation of Image Distortion by Atmospheric Turbulence Using Time-Series Image Data with 250-Million-Pixels. IJCEE, International

Journal of Computer Electrical Engineering, Vol. 10. 2018, ISSN 1793-8163

[61] Repasi E., Weiss R. Analysis of image distortions by atmospheric turbulence and computer simulation of turbulence effects. SPIE, Infrared Imaging Systems: Design, Analysis, Modeling, and Testing, Vol. 6941. 2008

[62] Gilles J., Dagobert T., De Franchis C. Atmospheric turbulence restoration by diffeomorphic image registration and blind deconvolution. Springer, Advanced Concepts for Intelligent Vision Systems, Vol. 5259. 2008, ISBN 978-3-540-88458-3

[63] Lau C.P., Lai Y.H., Lui L.M. Restoration of Atmospheric Turbulence-distorted Images via RPCA and Quasiconformal Maps. Computer Vision and Pattern Recognition. 2017

[64] Mao Y., Gilles J. Non Rigid Geometric Distortions Correction - Application To Atmospheric Turbulence Stabilization. Inverse Problems & Imaging, Vol. 6. 2012, ISSN 1930-8337

[65] Gajjar R., Zaveri T., Shukla A. Invariants based blur classification algorithm. IEEE, 5th Nirma University International Conference on Engineering. 2009

[66] P., THEUWISSEN A.J. Solid State Imaging with charge-Coupled Devices. Springer. 1995

[67] Janesick J.R. Scientific Charge-Coupled Devices. Spie Press Monograph. 2000

[68] Ishihara Y., Oda E., Tanigawa H., Teranishi N., Takeuchi E., Akiyama I. et al. Interline CCD image sensor with an anti-blooming structure, 1982 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, Vol. XXV 1982

[69] Svetkoff D.J. Image Quality Evaluation Of Machine Vision Sensors. SPIE, Optics, Illumination, and Image Sensing for Machine Vision II, Vol. 0850. 1988

[70] Nakamura J. Image sensors and signal processing for digital still cameras. Taylor & Francis. 2016

[71] Gong Y., Poellabauer C. An Overview of Vulnerabilities of Voice Controlled Systems. ArXiv. 2018, доступно по https://arxiv.org/pdf/1803.09156.pdf

[72] Giechaskiel I., Rasmussen K.B. Taxonomy and Challenges of Out-of-Band Signal Injection Attacks and Defenses. ArXiv. 2019, доступно по https://arxiv.org/pdf/1901.06935.pdf

[73] Vaidya T., Zhang Y., Sherr M., Shields C. Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition. USENIX Association, Proceedings of the 9th USENIX Conference on Offensive Technologies. 2019

[74] Carlini N., Mishra P., Vaidya T., Zhang Y., Sherr M., Shields C. et al. Hidden voice commands. USENIX Association, 2016, 25th USENIX Security Symposium. 2016, pp.513-530, ISBN 978-1-931971-32-4, доступно по https://nicholas.carlini.com/papers/2016_usenix_hiddenvoicecommands.pdf

[75] Alzantot M., Balaji B., Srivastava M. Did you hear that? Adversarial examples against automatic speech recognition. ArXiv. 2018, доступно по https://arxiv.org/pdf/1801.00554.pdf

[76] Taori R., Kamsetty A., Chu B., Vemuri N. Targeted adversarial examples for black box audio systems. ArXiv. 2018, доступно из: https://arxiv.org/pdf/1805.07820.pdf

[77] Du T., Ji S., Li J., Gu Q., Wang T., Beyah R. Sirenattack: Generating adversarial audio for end-to-end acoustic systems. ArXiv. 2019, доступно по https://arxiv.org/abs/1901.07846

[78] Carlini N., Wagner D. Audio adversarial examples: Targeted attacks on speech-to-text. IEEE Security and Privacy Workshops. 2018

[79] Qin Y., Carlini N., Goodfellow I., Cottrell G., Raffel C. Imperceptible, Robust and Targeted Adversarial Examples for Automatic Speech Recognition. Proceedings of the 36th International Conference on Machine Learning. 2019

[80] Schönherr L., Kohls K., Zeiler S., Holz T., Kolossa D. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. 26th Annual Network and Distributed System Security Symposium. 2018, доступно по https://arxiv.org/abs/1808.05665

[81] Neekhara P., Hussain S., Pandey P., Dubnov S., Mc Cauley J., Koushnafar F. Universal Adversarial Perturbations for Speech Recognition Systems. ArXiv. 2019, доступно по https://arxiv.org/abs/1905.03828

[82] Yuan X., Chen Y., Zhao Y., Long Y., Liu X., Chen K. et al. CommanderSong: A systematic approach for practical adversarial voice recognition. USENIX Association. 2018, 27th USENIX Security Symposium USENIX Security 18. 2018, pp.49-64, доступно по https://arxiv.org/abs/1801.08535

[83] Zhang G., Yan C., Ji X., Zhang T., Zhang T., Xu W. DolphinAttack: Inaudible Voice Commands. ACM, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, ISBN 978-1-4503-4946-8, доступно по https://arxiv.org/pdf/1708.09537.pdf

[84] Roy N., Shen S., Hassanieh H., Choudhury R.R. Inaudible voice commands: The long-range attack and defense. USENIX Association, 15th USENIX Symposium on Networked Systems Design and Implementation. 2018, ISBN 978-1-939133-01-4, доступно по https://synrg.csl.illinois.edu/papers/lipread_nsdi18.pdf

[85] Roy N., Hassanieh H., Roy Choudhury R. Backdoor: Making microphones hear inaudible sounds. ACM, Proceedings of the 15th Annual International Conference on Mobile Systems, Applications and Services. 2017, ISBN 978-1-4503-6661-8

[86] Song L.P., Mittal P. Inaudible voice commands. ArXiv. 2017, доступно по https://arxiv.org/abs/1708.07238

[87] Esmaeilpour M., Cardinal P., Koerich A.L. A Robust Approach for Securing Audio Classification Against Adversarial Attacks. ArXiv. 2019, доступно по https://arxiv.org/abs/1904.10990

[88] Winskel G., The formal semantics of programming languages: an introduction. MIT Press. 1993, ISBN 9780262231695

[89] Hoare C.A. An axiomatic basis for computer programming. ACM, Communications of the ACM, Vol. 12. 1969, ISSN 0001-0782

[90] Cousot P., Cousot R. Static determination of dynamic properties of programs. Proceedings of the Second International Symposium on Programming. 1976

[91] Logozzo F., Frahndrich M. Pentagons: a weakly relational abstract domain for the efficient validation of array accesses. ACL, Proceedings of the 2008 ACM symposium on Applied computing. 2008, ISBN 978-1-59593-753-7

[92] Mine A. The octagon abstract domain. Springer, Higher-Order and Symbolic Computation, Vol. 19. 2006

[93] Mukherjee R., Schrammel P., Haller L., Kroening D., Melham T. Lifting CDCL to Templatebased Abstract Domains for Program Verification. Springer, Automated Technology for Verification and Analysis. 2017, ISBN 978-3-319-68167-2

[94] Cousot P., Halbwachs N. Automatic Discovery of Linear Restraints Among Variables of a Program. ACM, Proceedings of the 5th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. 1978

[95] Goubault E., Le Gall T., Putot S. An Accurate Join for Zonotopes, Preserving Affine Input/Output Relations. Electronic Notes in Theoretical Computer Science, Vol. 287. 2012

[96] Specht F., Otto J., Niggemann O., Hammer B. Generation of Adversarial Examples to Prevent Misclassification of Deep Neural Network based Condition Monitoring Systems for Cyber-Physical Production Systems. IEEE, 16th International Conference on Industrial Informatics. 2018

[97] ISO/IEC 2382:2015 Information technology Vocabulary

[98] ISO/IEC/IEEE 15288:2015 Software and systems engineering Systems and software engineering System life cycle processes

[99] ISO/IEC 25000:2014 Systems and software engineering Systems and software Quality Requirements and Evaluation (SQuaRE) Guide to SQuaRE

[100] ISO/IEC/IEEE 26513:2017 Systems and software engineering Requirements for testers and reviewers of information for users

УДК 004.01:006.354 ОКС 35.020

Ключевые слова: информационные технологии, искусственный интеллект, робастность, оценка робастности, нейронные сети

Электронный текст документа
подготовлен АО "Кодекс" и сверен по:
официальное издание
М.: ФГБУ "РСТ", 2022