ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПНСТ 836— 2023 (ISO/IEC DTR 5469)

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Функциональная безопасность и системы искусственного интеллекта

(ISO/IEC DTR 5469, MOD)

Издание официальное

Москва
Российский институт стандартизации
2023

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 проекта стандарта, 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 29 ноября 2023 г. № 75-пнст

4 Настоящий стандарт является модифицированным по отношению к проекту международного стандарта ISO/IEC DTR 5469 «Искусственный интеллект. Функциональная безопасность и системы искусственного интеллекта» (ISO/IEC DTR 5469 «Artificial intelligence — Functional safety and AI systems», MOD) путем изменения отдельных ссылок, которые выделены в тексте курсивом, для учета особенностей российской национальной стандартизации.

При этом в него не включена таблица 3 примененного проекта международного стандарта, которую нецелесообразно применять в российской национальной стандартизации. Указанная таблица, не включенная в основную часть настоящего стандарта, приведена в дополнительном приложении ДА

Правила применения настоящего стандарта и проведения его мониторинга установлены в ГОСТ Р 1.16—2011 (разделы 5 и 6).

Федеральное агентство по техническому регулированию и метрологии собирает сведения о практическом применении настоящего стандарта. Данные сведения, а также замечания и предложения по содержанию стандарта можно направить не позднее чем за 4 мес до истечения срока его действия разработчику настоящего стандарта по адресу: info@tc164.ru и/или в Федеральное агентство по техническому регулированию и метрологии по адресу: 123112 Москва, Пресненская набережная, д. 10, стр. 2.

В случае отмены настоящего стандарта соответствующая информация будет опубликована в ежемесячном информационном указателе «Национальные стандарты» и также будет размещена на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.rst.gov.ru)

Содержание

Введение

Использование технологии ИИ значительно расширилось в последние годы и преимущество ИИ в некоторых сферах применения очевидно. Однако на данный момент имеется очень мало руководств по спецификации, проектированию, верификации функционально безопасных систем ИИ или по применению технологии ИИ для таких функций, которые связаны с безопасностью. Функции, реализуемые с помощью технологии ИИ, например МО, трудно объяснимы, и еще сложнее гарантировать их функционирование. Таким образом, где бы ни применялась технология ИИ, и особенно если она применяется для реализации систем, связанных с безопасностью, с большой долей вероятности возникают особые обстоятельства, требующие рассмотрения.

Доступность мощных технологий вычисления и хранения данных делает возможным применение МО в широких масштабах. Для все возрастающего числа сфер деятельности применение МО как технологии ИИ позволяет быстро и успешно разрабатывать алгоритмы, которые обнаруживают закономерности и тренды в данных. МО может также использоваться для идентификации аномального поведения или схождения функции на оптимальном решении в особой среде. МО с успехом применяется для анализа данных в финансовой сфере, в приложениях социальных сетей, для распознания речи и изображений (особенно при опознании личности), а также в управлении здравоохранением и прогнозировании, для создания цифровых ассистентов, промышленных роботов, для мониторинга состояния здоровья и в беспилотных транспортных средствах.

В дополнение к МО другие технологии ИИ приобретают важное значение в инженерных приложениях. Прикладная статистика, теория вероятности и теория статистического оценивания способствовали значительному прогрессу в сфере роботизации и распознавания образов. В результате технологии ИИ и системы ИИ начинают применяться в приложениях, которые имеют отношение к безопасности.

МО может использоваться для создания моделей, таким образом расширяя понимание окружающего мира. Но модели, полученные в результате МО, хороши лишь настолько, насколько качественной была информация, использованная при создании модели. Если обучающие данные не отражают достаточное число значимых сценариев применения, то полученные модели могут быть неверными. По мере поступления новых наблюдений они могут использоваться для подкрепления модели, но это, в свою очередь, может привести к необъективности в относительной важности наблюдений, что может сместить поведение алгоритма в сторону от менее частотных, но все-таки реальных типов поведения. Непрерывное дообучение и подкрепление модели новыми примерами помогает приблизить параметры модели к оптимальным значениям, но также возможен и побочный эффект: настройка преимущественно на типовые входные данные может снизить качество применения модели для экстремальных, хотя не менее важных, входных данных.

Цель настоящего стандарта заключается в том, чтобы дать возможность разработчику систем, связанных с безопасностью, применить надлежащим образом технологии ИИ как часть функций безопасности, используя понимание характеристик, факторов риска, существующих методов анализа функциональной безопасности и потенциальные ограничения технологий ИИ. В настоящем стандарте также приведена информация о проблемах и концепциях решений, связанных с функциональной безопасностью систем ИИ.

Целью настоящего стандарта не является установление требований. Описание требований к уровню полноты безопасности, позволяющих использовать элемент ИИ в функции безопасности с определенным уровнем полноты безопасности или уровень полноты автомобильной безопасности, находятся за пределами области применения настоящего стандарта.

В разделе 5 приведены общие положения функциональной безопасности и ее связи с технологией и системами ИИ.

В разделе 6 представлены разные классы технологии ИИ, чтобы показать их соответствие существующим стандартам функциональной безопасности в том случае, когда технология ИИ является частью функции безопасности. В разделе 6 также представлены разные уровни использования технологии ИИ в зависимости от их конечного воздействия на систему. Наконец, в разделе 6 приведена качественная оценка соответствующих уровней функциональной безопасности с разными сочетаниями класса технологии ИИ и уровня использования.

В разделе 7 предложен метод для использования технологии ИИ в системах, связанных с безопасностью, в которых соответствие существующим стандартам функциональной безопасности не может быть прямо продемонстрировано.

В разделе 8 приведены характеристики и связанные с ними факторы риска для функциональной безопасности в системах ИИ. Приведено описание проблем, связанных с использованием ИИ, и характеристик, которые могут быть приняты во внимание при попытке решить эти проблемы или ослабить их воздействие.

В разделах 9—11 приведены возможные решения указанных проблем в области верификации и валидации, управления и защитных мер, процессов и методик.

В приложениях приведены примеры применения настоящего стандарта и дополнительная информация. В приложении А показано, как *ГОСТ IEC 61508-3* применяется к элементам технологии ИИ. В приложении В приведен пример применения принципа трехступенчатой реализации и определения разнообразных характеристик. В приложении С дано детальное описание процессов, описанных в 9.3. В приложении D показана связь между жизненным циклом безопасности, установленным в *ГОСТ Р МЭК 61508-1*, и жизненным циклом системы ИИ, установленным в [1].

Федеральное агентство
по техническому регулированию
и метрологии
Федеральное агентство
по техническому регулированию
и метрологии

ПНСТ 836—2023
(ISO/IEC DTR 5469)

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Функциональная безопасность и системы искусственного интеллекта

Artificial intelligence.
Functional safety and AI systems

Срок действия — с 2024—01—01
до 2027—01—01

1 Область применения

В настоящем стандарте приведено описание свойств, соответствующих факторов риска, существующих методов и процессов, связанных с применением:

- ИИ внутри функции, связанной с обеспечением безопасности, для реализации соответствующего функционала;
- функций обеспечения безопасности без использования ИИ для обеспечения безопасности оборудования, управляемого ИИ;
- систем ИИ для проектирования и создания функций обеспечения безопасности.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ IEC 61508-3—2018 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 3. Требования к программному обеспечению

ГОСТ ISO 12100 Безопасность машин. Основные принципы конструирования. Оценки риска и снижения риска

ГОСТ Р 59276—2020 Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения

ГОСТ Р 70462.1—2021/ISO/IEC TR 24029-1—2021 Информационные технологии. Интеллект искусственный. Оценка робастности нейронных сетей. Часть 1. Обзор

ГОСТ Р ИСО 26262-1 Дорожные транспортные средства. Функциональная безопасность. Часть 1. Термины и определения

ГОСТ Р ИСО 26262-5 Дорожные транспортные средства. Функциональная безопасность. Часть 5. Разработка аппаратных средств изделия

ГОСТ Р ИСО 26262-6—2021 Дорожные транспортные средства. Функциональная безопасность. Часть 6. Разработка программного обеспечения изделия

ГОСТ Р ИСО/МЭК 18045 Информационная технология. Методы и средства обеспечения безопасности. Методология оценки безопасности информационных технологий

ГОСТ Р ИСО/МЭК ТО 19791 Информационная технология. Методы и средства обеспечения безопасности. Оценка безопасности автоматизированных систем

ГОСТ Р ИСО/МЭК 25010 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

Издание официальное

ГОСТ Р ИСО/МЭК 27001 Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования

ГОСТ Р МЭК 61508-1—2012 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 1. Общие требования

ГОСТ Р МЭК 61508-2 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 2. Требования к системам

ГОСТ Р МЭК 61508-4—2012 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 4. Термины и определения

ГОСТ Р МЭК 61508-7—2012 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 7. Методы и средства

ГОСТ Р МЭК 61511-1 Безопасность функциональная. Системы безопасности приборные для промышленных процессов. Часть 1. Термины, определения и технические требования

ГОСТ Р МЭК 62443-2-1 Сети коммуникационные промышленные. Защищенность (кибербезопасность) сети и системы. Часть 2-1. Составление программы обеспечения защищенности (кибербезопасности) системы управления и промышленной автоматики

ГОСТ Р МЭК 62443-3-3 Сети промышленной коммуникации. Безопасность сетей и систем. Часть 3-3. Требования к системной безопасности и уровни безопасности

ПНСТ 839—2023 (ISO/IEC TR 24027:2021) Искусственный интеллект. Смещенность в системах искусственного интеллекта и при принятии решений с помощью искусственного интеллекта

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применены термины по *[2]*, а также следующие термины с соответствующими определениями.

3.1

безопасность (safety): Отсутствие неприемлемого риска.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.1.11*]

3.2

функциональная безопасность (functional safety): Часть общей безопасности, обусловленная применением УО и системы управления УО и зависящая от правильности функционирования Э/Э/ПЭ систем, связанных с безопасностью, и других средств по снижению риска.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.1.12*]

3.3

риск; риск функциональной безопасности (risk; functional safety risk): Сочетание вероятности события причинения вреда и тяжести этого вреда.

Примечание — Дальнейшее обсуждение этого определения содержится в МЭК 61508-5, приложение А.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.1.6*]

3.4

риск; организационный риск (risk; organizational risk): Следствие влияния неопределенности на достижение поставленной цели.

Примечание 1 — Под следствием влияния неопределенности необходимо понимать отклонение от ожидаемого результата или события (позитивное и/или негативное).

Примечание 2 — Цели могут быть различными по содержанию (в области экономики, здоровья, экологии и т. п.) и назначению (стратегические, общеорганизационные, относящиеся к разработке проекта, конкретной продукции и процессу).

Примечание 3 — Риск часто характеризуют путем описания возможного события (3.5) и его последствий (3.6) или их сочетания.

Примечание 4 — Это общее определение риска. Поскольку риски рассматриваются узко с точки зрения вреда (3.5), специализированное значение термина «риск» (3.3) используется в настоящем стандарте в дополнение к общему определению термина.

[*Адаптировано из ГОСТ Р ИСО 31000—2019, пункт 3.1*]

3.5

вред (harm): Физические повреждения или ущерб, причиняемый здоровью людей, имуществу или окружающей среде.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.1.1*]

3.6

опасность (hazard): Потенциальный источник причинения вреда.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.1.2*]

3.7

опасное событие (hazardous event): Событие, в результате которого может быть причинен вред.

Примечание — Причинение вреда в результате опасного события зависит от того, подвергаются ли люди, имущество или окружающая среда воздействию последствий опасного события, и если причинение людям вреда возможно, то могут ли они избежать последствий события после того, как оно произошло.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.1.4*]

3.8

система (system): Комбинация взаимодействующих элементов, организованных для достижения одной или нескольких поставленных целей.

[*Адаптировано из ГОСТ Р 57193—2016, пункт 4.1.44*]

3.9

систематический отказ (systematic failure): Отказ, связанный детерминированным образом с какой-либо причиной, которая может быть исключена только путем модификации проекта либо производственного процесса, операций, документации, либо других факторов.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.6.6*]

3.10

система, связанная с безопасностью (safety-related system): Система, которая:

- реализует необходимые функции безопасности, требующиеся для достижения и поддержки безопасного состояния УО, и
- предназначена для достижения своими средствами или в сочетании с другими Э/Э/ПЭ системами, связанными с безопасностью, и другими средствами снижения риска необходимой полноты безопасности для требуемых функций безопасности.

[*Адаптировано из ГОСТ Р МЭК 61508-4—2012, статья 3.4.1*]

3.11

функция безопасности (safety function): Функция, реализуемая Э/Э/ПЭ системой, связанной с безопасностью, или другими мерами по снижению риска, предназначенная для достижения или поддержания безопасного состояния УО по отношению к конкретному опасному событию (3.7).

[*Адаптировано из ГОСТ Р МЭК 61508-4—2012, статья 3.5.1*]

3.12

управляемое оборудование; УО (equipment under control; EUC): Оборудование, машины, аппараты или установки, используемые для производства, обработки, транспортирования, в медицине или в других процессах.

Примечание — «Системы управления УО» представляют собой отдельное, отличное от УО, понятие.

[*ГОСТ Р МЭК 61508-4—2012, статья 3.2.1*]

3.13

программируемая электроника; ПЭ (programmable electronic; PE): Средство, основанное на использовании компьютерных технологий и которое может включать в себя аппаратное и программное обеспечение, а также устройства ввода и/или вывода.

Примечание — Данный термин охватывает микроэлектронные устройства, основанные на одном или нескольких центральных процессорах (ЦП) и связанных с ними устройствах памяти и т. п.

Пример — К программируемым электронным устройствам относятся:
- микропроцессоры;
- микроконтроллеры;
- программируемые контроллеры;
- специализированные интегральные схемы;
- программируемые логические контроллеры;
- другие устройства на основе компьютерных технологий (например, микропроцессорные датчики, преобразователи, устройства привода).

[*ГОСТ Р МЭК 61508-4—2012, статья 3.2.12*]

3.14

электрический/электронный/программируемый электронный; Э/Э/ПЭ (electrical/electronic/programmable electronic; E/E/PE): Основанный на электрической (Э), и/или электронной (Э), и/или программируемой электронной (ПЭ) технологии.

Примечание — Данный термин предназначен для того, чтобы охватить любое или все устройства или системы, действующие на основе электричества.

Пример — В число электрических/электронных/программируемых электронных устройств входят:
- электромеханические устройства (электрические);
- твердотельные непрограммируемые электронные устройства (электронные);
- электронные устройства, основанные на компьютерных технологиях (программируемые электронные).

[*ГОСТ Р МЭК 61508-4—2012, статья 3.2.13*]

3.15 **технология ИИ** (AI technology): Технология, используемая для применения модели ИИ (3.16).

3.16 **модель ИИ** (AI model): Физическое, математическое или иное логическое представление системы, объекта, явления, процесса или данных.

Примечание — См. [2], пункт 3.1.23.

3.17 **тестовый оракул** (test oracle): Источник информации для определения успешности прохождения теста.

Примечание — См. [3], пункт 3.115.

4 Сокращения

В настоящем стандарте применены следующие сокращения:

ИИ — искусственный интеллект;
ИНС — искусственная нейронная сеть;
ИТ — информационные технологии;
КПЭ — ключевой показатель эффективности;
МО — машинное обучение;
ПО — программное обеспечение;
СНС — сверточная нейронная сеть;
CUDA — архитектура унифицированного вычислительного устройства (compute unified device architecture);
FMEA — анализ видов и последствий отказов (failure modes and effects analysis);
JPEG — объединенная группа экспертов по фотографии (joint photographic experts group).

5 Общие положения функциональной безопасности

5.1 Введение в функциональную безопасность

Сфера функциональной безопасности фокусируется на рисках, связанных с травмами и вредом для здоровья людей, окружающей среды и, в некоторых случаях, снижением вреда для продукции или оборудования. Определение риска различается в зависимости от области применения термина, см. раздел 3. Оба приведенных определения являются важными понятиями для использования ИИ. Все упоминания понятия «риск» далее в настоящем стандарте относятся к определению с точки зрения функциональной безопасности.

В соответствии с *ГОСТ Р МЭК 61508-1* управление риском является итерационным процессом оценки и снижения риска. Оценка риска выявляет источник вреда и риски, связанные с целевым использованием и с разумно предсказуемым неправильным использованием продукта или системы. Снижение риска уменьшает риски до уровня, который становится допустимым. Допустимым риском считается такой уровень риска, который приемлем в заданном контексте при существующем уровне развития технологии.

Серия стандартов *ГОСТ Р МЭК 61508* в качестве наилучшей практики устанавливает следующий подход, состоящий из трех стадий:

- стадия 1: функционально безопасный проект;
- стадия 2: ограничительные и защитные устройства;
- стадия 3: информация для конечного пользователя.

Снижение риска путем обеспечения функциональной безопасности относится к стадии 2.

Настоящий стандарт ограничивается рассмотрением аспектов функций безопасности, выполняемых системой, связанной с безопасностью, которая использует технологии ИИ, либо в рамках системы, связанной с безопасностью, либо в процессе проектирования и создания системы, связанной с безопасностью (стадия 2).

В настоящем стандарте не рассматривается методика использования технологии ИИ на стадиях 1 и 3.

5.2 Функциональная безопасность

ГОСТ Р МЭК 61508-4 определяет функциональную безопасность как «часть общей безопасности, обусловленную применением УО и системы управления УО и зависящую от правильности функционирования Э/Э/ПЭ систем, связанных с безопасностью, и других средств по снижению риска». Э/Э/ПЭ система, связанная с безопасностью, обеспечивает функцию безопасности, которая в *ГОСТ Р МЭК 61508-4* определена как «функция, реализуемая Э/Э/ПЭ системой, связанной с безопасностью, или другими мерами по снижению риска, предназначенная для достижения или поддержания безопасного состояния УО по отношению к конкретному опасному событию». Иными словами, функции безопасности управляют риском, связанным с опасностью, которая может нанести вред людям или окружающей среде. Функции безопасности могут также снизить риск серьезных экономических последствий.

Как следует из термина, функциональная безопасность, как она определена по *ГОСТ Р МЭК 61508-4*, направлена на достижение и поддержание функционально безопасного состояния УО путем обеспечения функций безопасности. Исходя из включения в определение функциональной безопасности и функций безопасности формулировки «другие меры по снижению риска» нетехнические функции также явно учтены. Понятие УО не ограничивается отдельными устройствами, оно может быть также и системами.

В соответствии с приведенными определениями функциональная безопасность как сфера деятельности связана, таким образом, с надлежащим выстраиванием этих собственно технических и нетехнических функций безопасности для снижения риска или ограничения уровня риска для конкретного УО, начиная с уровня компонентов и заканчивая уровнем системы в целом, принимая во внимание человеческий фактор и учитывая условия эксплуатации и воздействие окружающей среды.

Функциональная безопасность сфокусирована на функциях безопасности для снижения риска и на свойствах этих функций, требуемых для снижения риска. Хотя функционал любой функции безопасности привязан к конкретному контексту ее использования, те свойства, которые требуются для снижения риска, и связанные с ними меры находятся в центре внимания при стандартизации функциональной безопасности.

До появления программируемых систем, когда функции безопасности были ограничены в применении аппаратными средствами, основной целью функциональной безопасности была минимизация последствий и вероятности случайных отказов оборудования. При все большей роли ПО в применении функций безопасности фокус внимания сдвинулся в сторону систематических отказов на этапах проектирования и создания.

Примечание — *[4]* описывает требования к безопасности для целевого функционала, включая такие аспекты, как ограничение характеристик. В приложении D приведено описание их интерпретации для МО.

Накоплена база знаний о том, как избежать систематических отказов в системах, не использующих ИИ, и при разработке ПО. Настоящий стандарт рассматривает использование ИИ в контексте функций безопасности. Функции, включающие технологии ИИ, особенно МО, обычно разрабатываются иначе, чем системы без ИИ. Они менее привязаны к спецификациям и в большей степени опираются на отслеживание данных, определяющих поведение системы. Именно поэтому каталог существующих мер предотвращения систематических отказов дополняется, принимая во внимание особенности технологий ИИ: в приложении А приведен пример такого дополнения. Меры по снижению рисков, характерные для ИИ, также отличаются от мер, применяемых для систем без ИИ, с функциональной точки зрения. Функциональная безопасность ставит во главу угла стойкость к систематическим отказам по *ГОСТ Р МЭК 61508-4—2012* (статья 3.5.9) в дополнение к случайным отказам оборудования и систематическим отказам на протяжении всего жизненного цикла.

Предназначение технологии ИИ при реализации функции безопасности состоит в том, что они дают возможность обратиться к новым методам снижения риска. Настоящий стандарт посвящен исследованию применения технологии ИИ для этих целей, не пренебрегая существующими концепциями снижения риска, путем предложения некоторых положений, связанных с риском и классификацией.

В целом достижение приемлемого уровня риска для систем, уровень сложности и автоматизации которых все более возрастает, вероятно, связано с усовершенствованием концепций безопасности. Оно связано с адекватным применением как технических, так и иных мер снижения риска для достижения и поддержания системы в безопасном состоянии. Гарантировать валидность таких усовершенствованных концепций безопасности — большой вызов в области функциональной безопасности. Данный вызов приводит к возрастанию числа требований функциональной безопасности. Для всех технических мер снижения риска отличительной особенностью является учет случайных и систематических отказов оборудования, приведенных в серии стандартов *ГОСТ Р МЭК 61508* и в стандартах, разработанных на их основе. Однако в случае функций безопасности на основе технологии ИИ неизбежно потребуется дополнительный фокус внимания на необходимости обеспечения гарантии в отношении того, что возможности систем, которые используют эти функции, достаточны для той среды, в которой они будут применяться.

6 Использование технологии ИИ в Э/Э/ПЭ системах, связанных с безопасностью

6.1 Постановка проблемы

Использование технологии и систем ИИ на момент разработки настоящего стандарта не описано ни в одном стандарте функциональной безопасности, а в некоторых стандартах их использование запрещено. Стандарты, в которых упоминается ИИ: *[4]*, *[5]* и *[6]*.

6.2 Технологии ИИ в Э/Э/ПЭ системах, связанных с безопасностью

Э/Э/ПЭ системы, связанные с безопасностью, имеют набор характеристик, которые надежно обеспечивают заданный уровень безопасности. В идеальном случае эти характеристики универсальны и не зависят от условий применения в конкретном случае. Однако данные и спецификации варьируются в зависимости от конкретного случая применения и сферы технологий. Процесс определения характеристик показан на рисунке 3 (см. 7.2) для каждой из трех стадий фреймворка ИИ. Характеристики могут определяться для каждого отдельного случая, относящегося к конкретному применению и сфере технологии, связанными с ними данными и спецификациями. Некоторые из этих характеристик могут быть основаны на таких стандартах, как *ГОСТ IEC 61508-3, ГОСТ Р ИСО 26262-1, ГОСТ Р ИСО 26262-5, ГОСТ Р ИСО 26262-6, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ Р МЭК 61508-4, [7], [8], [9], [10].* Другие характеристики определены недавно. В разделе 8 приведен перечень типичных характеристик.

Соответствие заданным характеристикам требует выполнения конкретных требований функциональной безопасности при реализации, установке, валидации таких систем, а также при управлении ими и их техобслуживании. Например, *ГОСТ IEC 61508-3* определяет требования для ПО Э/Э/ПЭ систем. Но некоторые технологии ИИ используют иные подходы на стадии создания системы (например, МО), чем те, которые использовались до применения ИИ в разработках, на которые распространяется *ГОСТ IEC 61508-3.*

Для рассмотрения отличий традиционных подходов разработчиков и подхода, типичного для использования ИИ, в настоящем подразделе приведена общая схема классификации, которая показывает применимость технологии ИИ для Э/Э/ПЭ систем, связанных с безопасностью, в разных контекстах применения технологии ИИ.

Пример схемы классификации приведен в таблице 1, а соответствующая блок-схема — на рисунке 1. На схеме показано, как именно технология ИИ может применяться в контексте функциональной безопасности для конкретного применения. Соответствующие отраслевые стандарты применяются для конкретизации общей схемы классификации в конкретные практические требования.

Схема классификации (таблица 1) организована согласно двум признакам:

1) Применение ИИ и уровень использования. Этот признак учитывает применение ИИ и способы его использования. Он классифицируется от A до D, с промежуточными уровнями для A и B.

Примечание — Факторы, приведенные в разделе 8, важны в контексте классификации. Они включают уровень автоматизации и управления (8.2), степень понятности и объяснимости (8.3), проблемы, связанные с окружающей средой (8.4), устойчивость к состязательным вводам данных и злонамеренному воздействию (8.5), проблемы аппаратных средств, связанные с применением ИИ (8.6), и зрелость технологии (8.7).

Пример классификации уровня использования:

- уровень использования A1 присваивается, когда технология ИИ используется в Э/Э/ПЭ системе, связанной с безопасностью, где возможно автоматическое принятие решения системной функции с использованием технологии ИИ;
- уровень использования A2 присваивается, когда технология ИИ используется в Э/Э/ПЭ системе, связанной с безопасностью, где невозможно автоматическое принятие решения системной функции с использованием технологии ИИ (например, ИИ используется в Э/Э/ПЭ системе для функции диагностики).

Примечание — Оценка может меняться в зависимости от роли функции диагностики, например от того, насколько диагностика критична для поддержания функциональной безопасности системы, или она является лишь одним из множества факторов функциональной безопасности наряду с другими;

- уровень использования B1 присваивается, когда технология ИИ используется только на стадии создания Э/Э/ПЭ системы, связанной с безопасностью (например, как вспомогательный инструмент

вне сети), и возможно автоматическое принятие решения разработанной функцией с использованием технологии ИИ;

- уровень использования B2 присваивается, когда технология ИИ используется только на стадии создания Э/Э/ПЭ системы, связанной с безопасностью (например, как вспомогательный инструмент вне сети), и невозможно автоматическое принятие решения разработанной функцией;

- уровень использования C присваивается, когда технология ИИ не является частью функции функциональной безопасности в Э/Э/ПЭ системе, но может иметь опосредованное влияние на эту функцию.

Примечание — Уровень использования C относится к применению ИИ, которое явно обеспечивает дополнительное снижение риска, при этом его отказ не является критичным для уровня приемлемого риска.

Пример — Применение ИИ для увеличения или снижения объема потребления в системе безопасности;

- уровень использования D присваивается, когда технология ИИ не является частью функции функциональной безопасности в Э/Э/ПЭ системе и не может иметь влияния на эту функцию из-за ее отделения и управления состоянием.

Примечание — Примером может быть разделение через изолированную программную среду или гипервизор таким образом, чтобы исключить влияние на функционал безопасности.

2) Класс технологии ИИ. Этот признак связан с уровнем реализации технологии ИИ при удовлетворении определенному набору характеристик:

- класс I присваивается, если технология ИИ может быть создана и проверена на основе существующих стандартов функциональной безопасности, например если характеристики и набор методов и приемов для достижения этих характеристик могут быть определены на основе существующих стандартов функциональной безопасности;

- класс II присваивается, если технология ИИ не может быть полностью создана и проверена на основе существующих стандартов функциональной безопасности, но все-таки возможно (как показано на рисунке 1) определить дополнительные требования, методы и приемы для разработки, создания, верификации и валидации желаемых характеристик безопасности для достижения необходимого снижения риска;

- класс III присваивается, если технология ИИ не может быть создана и проверена на основе существующих стандартов функциональной безопасности и при этом также невозможно удовлетворить все заданные требования по характеристикам на основе соответствующих методов и приемов.

Компоненты, включающие технологию ИИ, состоят из разнообразных элементов (см. раздел 7). Каждый элемент может принадлежать к отдельному классу технологии ИИ. Например, нижние уровни абстракции в нейронной сети могут применять библиотеки C++, которые сами по себе могут систематически подвергаться проверке (например, в соответствии с *ГОСТ IEC 61508-3*), см. приложение А. Как таковые, они могут быть отнесены к классу I, хотя более высокие уровни абстракции (например, модели глубокого обучения) будут скорее отнесены к классу II или III.

Для компонентов ИИ класса I применение существующих стандартов функциональной безопасности возможно и в целом желательно. Для компонентов ИИ класса II применение существующих стандартов функциональной безопасности приведет к достижению характеристик, требуемых для функциональной безопасности, только частично.

После этого к оставшимся параметрам применяются дополнительные методы и приемы, такие как дополнительная верификация и валидация (см. раздел 9). Эффективность дополнительных методов и приемов может быть различной для разных приложений и уровней использования. Для компонентов ИИ класса III на момент разработки настоящего стандарта не существует известных методов и приемов для идентификации набора характеристик, которые обеспечивают существенное снижение риска.

Примечание — Приведение подробного руководства по эффективности применения дополнительных методов и приемов для каждого приложения и уровня использования не является целью настоящего стандарта.

Таблица 1 — Пример таблицы классификации ИИ

<table>
<tr><th>Класс технологии ИИ. Применение и уровень использования ИИ</th><th>Класс I технологии ИИ</th><th>Класс II технологии ИИ</th><th>Класс III технологии ИИ</th></tr>
<tr><td>Уровень использования A1[a]</td><td rowspan="5">Возможно применение концепций снижения риска на основе существующих стандартов функциональной безопасности</td><td>Соответствующий набор требований[c]</td><td rowspan="5">На момент разработки настоящего стандарта не было известно о существовании подходящего набора свойств с соответствующими методами и приемами, обеспечивающими существенное снижение риска</td></tr>
<tr><td>Уровень использования A2[a]</td><td>Соответствующий набор требований[c]</td></tr>
<tr><td>Уровень использования B1[a]</td><td>Соответствующий набор требований[c]</td></tr>
<tr><td>Уровень использования B2[a]</td><td>Соответствующий набор требований[c]</td></tr>
<tr><td>Уровень использования C[a]</td><td>Соответствующий набор требований[c]</td></tr>
<tr><td>Уровень использования D[b]</td><td colspan="3">Нет конкретных требований по функциональной безопасности для технологии ИИ, но применяются концепции снижения риска из существующих стандартов функциональной безопасности</td></tr>
</table>

[a] Статическое (вне сети) обучение на стадии создания.

[b] Динамическое (в сети) обучение возможно.

[c] Соответствующий набор требований для каждого уровня использования может определяться путем применения концепций снижения риска в соответствии с существующими стандартами функциональной безопасности и дополнительных факторов, приведенных в разделах 8—11. Примеры приведены в приложении В. Определение детальных требований для каждого уровня использования не является задачей настоящего стандарта.

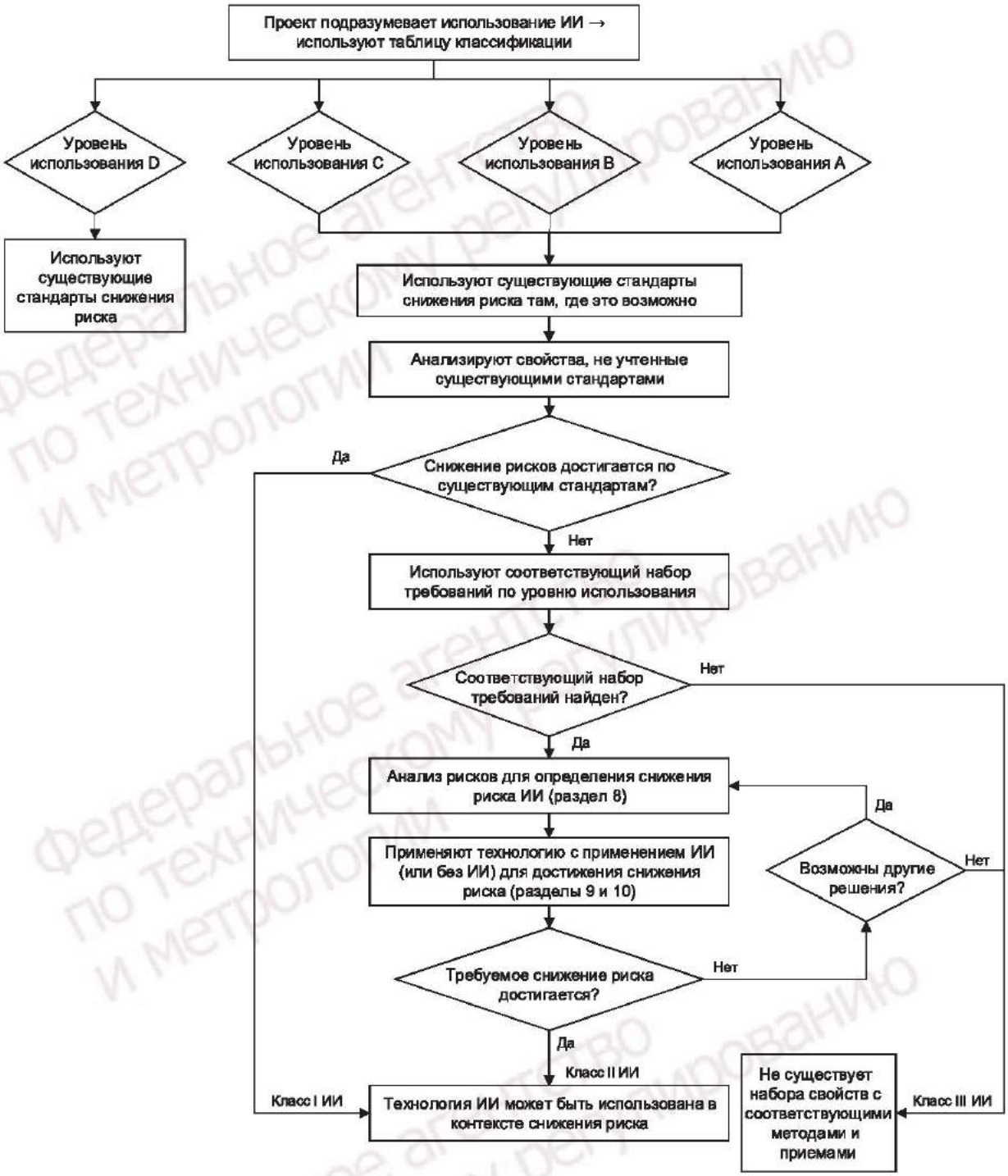


Рисунок 1 — Блок-схема для определения классификации технологий ИИ

7 Элементы технологии ИИ и принцип трех стадий ее реализации

7.1 Элементы технологии для создания и выполнения моделей ИИ

Создание и выполнение модели ИИ охватывает разные элементы технологии ИИ. В таблице 2 приведено высокоуровневое описание систем ИИ и задействованных типичных элементов исходя из функциональных слоев экосистемы ИИ в соответствии с *[2]*, рисунок 6.

Таблица 2 — Элементы технологии ИИ *[2]*

Элемент технологии ИИ
Интеллектуальные службы
МО: - создание модели и ее использование; - инструменты; - данные для МО
Проектирование: - знания из практического опыта в специальной области; - инструменты
Облачные и периферийные вычисления, большие данные, источники данных
Пул ресурсов — обработка, хранение, создание сети
Управление ресурсами — обеспечение доступа к ресурсам

К некоторым элементам технологии можно применить существующие концепции функциональной безопасности (например, программы, компилирующие модель в исполняемую программу). Однако все элементы технологии, участвующие в создании и выполнении модели, могут иметь отношение к соображениям безопасности, включая те, к которым можно применить существующие концепции функциональной безопасности, и те, для которых могут быть определены новые концепции. В приложении А приведен пример того, как существующие концепции функциональной безопасности могут применяться к технологии ИИ путем оценки применимости *ГОСТ IEC 61508-3*. В приложении В приведен пример того, как конкретные характеристики, такие как описанные в разделе 8, могут применяться к технологии ИИ, для которой существующие концепции функциональной безопасности применить невозможно.

Как показано на рисунке 2, элементы, включающие технологии ИИ, используются на разных уровнях системы или приложения:

- для элементов высокого уровня (граф приложения, модель МО и связанные с ними инструменты) могут быть применимы специфические характеристики, приведенные в разделе 8;
- для элементов более низкого уровня (программы в машинном коде и подобные элементы) применимы характеристики, не связанные с ИИ, приведенные в существующих стандартах, например *ГОСТ IEC 61508-3, ГОСТ Р ИСО 26262-1, ГОСТ Р ИСО 26262-5, ГОСТ Р ИСО 26262-6, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ Р МЭК 61508-4, [9]*.

Рисунок 2 — Иерархия элементов технологии (на примере МО)

7.2 Трехступенчатый принцип реализации системы ИИ

Любая система ИИ может быть представлена (см. рисунок 3) на основе принципа реализации в трех стадиях:

- сбор данных;
- индуцирование знаний из данных и человеческого знания;
- обработка и получение данных на выходе.

Примечания

1 В соответствии с *[2]*, рисунок 5, первая стадия связана с задачей ввода, вторая — с задачей обучения, третья — с задачей обработки данных.

2 В данном контексте, человеческое знание получается из самых разных источников, как в соответствующих сферах деятельности, так и в системах ИИ.

3 Предложенный принцип реализации является общим. Специфические более детальные примеры систем ИИ: отслеживание — анализ — план — исполнение (MAPE), ощущение — понимание — решение — действие (SUDA).

4 Цель трехступенчатого принципа реализации — не описание жизненного цикла (который описан в разделе 11 и включает все стадии: от разработки концепции и ее созревания до разработки требований), а демонстрация другой точки зрения.

5 Рисунок 3 не показывает петли обратной связи, которые могут быть применимы к системам ИИ, встроенным в контуры принятия решений или влияющим на ситуацию в реальном мире.

6 Индукция знаний включает обучение модели, в то время как обработка информации включает логический вывод.

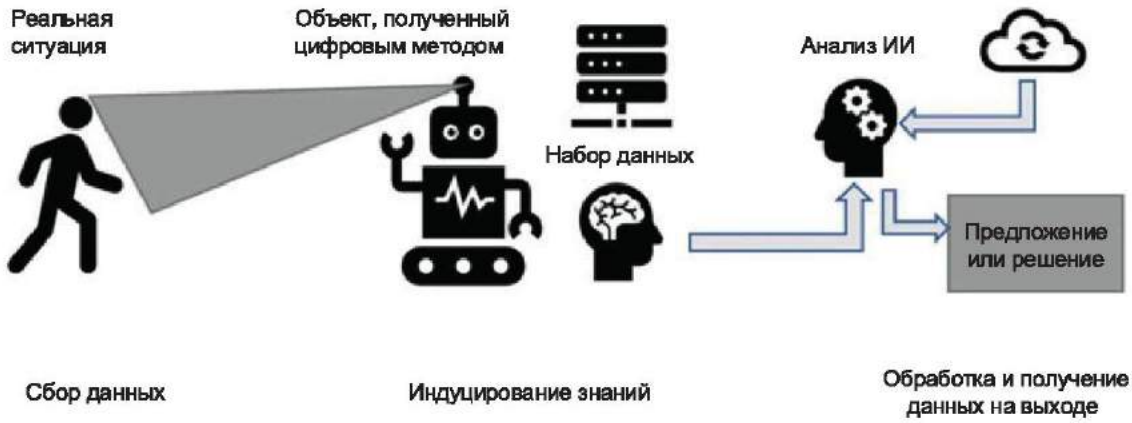


Рисунок 3 — Трехступенчатый принцип реализации

7.3 Вывод критериев приемлемости для трех стадий принципа реализации

Следующий процесс (см. рисунок 4) может быть применен для получения критериев приемлемости на основе трехступенчатого принципа реализации:

- желаемые характеристики определяются для каждой из трех стадий;
- характеристики связываются с темами и с конкретными методами и приемами, применимыми для этих тем;
- критерии приемлемости получаются из набора конкретных методов и приемов. Как показано на рисунке 4, критерии для приемлемости выбранных методов и приемов и, возможно, показателей и пороговых значений, таких как пределы оцениваемых неопределенностей, встроены в общее суждение о приемлемости. Общее суждение о приемлемости показывает, что выбранные критерии приемлемости гармонизированы с независимыми от технологии критериями приемлемости риска, такими как ALARP, MEM, GAMAB или PRB. Общее суждение о приемлемости может быть представлено, например, для гарантийного случая.

Рисунок 4 — Процессы на каждом этапе

Примечания

1 Характеристики могут определяться для каждого отдельного случая или выводиться из тех, которые указаны в существующих стандартах, в зависимости от уровня элементов, использующих технологию ИИ. В разделе 8 представлен список рассматриваемых характеристик.

2 «Желаемые характеристики», «темы, относящиеся к характеристикам» и «конкретные методы и приемы» могут быть индивидуальными для каждого этапа или общими, в зависимости от конкретного применения.

3 В данном контексте критерии приемлемости рассматриваются как то, что может быть выявлено и подтверждено в процессе разработки.

8 Характеристики систем ИИ и связанные с ними факторы риска

8.1 Введение

8.1.1 Общие положения

В разделе 7 описано, каким образом определение искомых характеристик становится первым этапом трехступенчатого принципа реализации. Характеристики привязаны к темам и в конечном счете к конкретным методам и приемам, применяемым для этих тем. Критерии приемлемости выделяются на основе конкретных методов и приемов.

Настоящий раздел является руководством, посвященным характеристикам, которые описывают системы, использующие технологии ИИ, и связанные с ними факторы риска. Такие характеристики и факторы риска включают уровень автоматизации и контроля (см. 8.2), степень понятности и объяснимости (см. 8.3), проблемы, связанные с окружающей средой (см. 8.4), устойчивость к состязательным вводам данных и злонамеренному воздействию (см. 8.5), проблемы аппаратных средств, связанные с применением ИИ (см. 8.6) и зрелость технологии (см. 8.7).

Характеристики систем, использующих технологию ИИ, и факторы риска подробно описаны в настоящем разделе.

8.1.2 Алгоритмы и модели

На технологическом уровне функциональные возможности ИИ часто определяются сочетанием модели ИИ и параметров этой модели. Параметры, которые генерирует алгоритм в процессе обучения, обычно отражают ту информацию, которая соответствует цели применения (например, как разные входные данные нужно различать и распознавать), в то время как модель вычисляет информацию на основе параметров и входных данных (например, для прогнозирования). Это значит, что функциональная безопасность приложения зависит и от того, и от другого.

Приведенные примеры типов моделей ИИ включают линейные функции, логические исчисления, динамические байесовские сети и ИНС. Параметры моделей могут быть установлены «вручную» инженером или синтезированы на основе данных с помощью алгоритмов МО, которые сами используют процесс систематического анализа. Модели обычно внедряются в виде, готовом к исполнению, в таком, как машинный код (для ПО) или специальные аппаратные средства, такие как программируемая пользователем вентильная матрица (field programmable gate arrays, FPGA).

Обычно модели сами по себе, без параметров, содержат ограниченный объем знаний или импликаций о предназначении приложения. Это примерно то же, что библиотеки базовых программ или программная среда (компиляторы и т. п.), используемые в ПО, не связанном с использованием ИИ. Правильное функционирование модели критически важно для обеспечения функциональной безопасности. Таким образом, целостность реализаций модели в сфере ИИ часто может регулироваться с помощью существующих принципов функциональной безопасности, приведенных, например, в *ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ Р МЭК 61508-4,* точно так же, как это происходит в случае программных компонентов, не использующих ИИ. То же применимо и к логике, задействованной в подготовке моделей и параметров к исполнению.

В противоположность вышесказанному, параметры модели часто содержат знания о цели применения систем, связанных с функциональной безопасностью. Существует несколько разных способов конструирования параметров и могут использоваться разные подходы к оценке полноты мер по снижению риска для обеспечения функциональной безопасности.

Например, если параметры модели создаются вручную инженерами, модели с большой вероятностью отражают знания инженеров о приложении, которое может быть оценено в процессе управления в рамках жизненного цикла функциональной безопасности. В таком случае можно использовать жизненный цикл из существующих стандартов (технология ИИ класса I — по 6.2). Часто можно легко создать параметры вручную для простых моделей, таких как простые линейные функции или логические исчисления.

В отдельных случаях параметры, полученные из данных с помощью алгоритмов МО, можно проанализировать и верифицировать после их получения. В других случаях параметры, полученные с помощью алгоритмов МО, можно извлечь и использовать их затем для обогащения знаний, которые могут использоваться, в свою очередь, для развития технологии ИИ в дальнейшем. При применении валидированного инженерного знания также может использоваться жизненный цикл из существующих стандартов функциональной безопасности (например, можно рассматривать эти модели как технологию ИИ класса I — по 6.2).

В других случаях параметры, полученные с помощью алгоритмов МО, могут быть слишком сложными для понимания, анализа и верификации. Это относится к сложным типам моделей, таким как нейросети, потому что представления моделей в этих случаях не всегда отражают человеческое понимание и способ рассуждения. Применение других подходов к оценке снижения риска для функциональной безопасности оправдано в таких случаях, и это может оказаться огромной проблемой при использовании технологии ИИ для создания систем функциональной безопасности.

8.2 Уровень автоматизации и контроля

Уровень автоматизации (называемый иногда «степень независимости») отражает степень, до которой система ИИ функционирует независимо, без надзора и управления со стороны человека. Эта характеристика определяет не только объем информации о системе, доступный оператору, но и варианты управляющих действий и вмешательства со стороны человека.

В рамках данной темы рассматривается вопрос о том, насколько высок уровень автоматизации для конкретного приложения, а также степень ограничения пользователя в выборе управляющих действий. Системы, использующие технологии ИИ с высокой степенью автоматизации, могут демонстрировать неожиданное поведение, которое трудно заметить и проконтролировать. Поэтому применение

высокоавтоматизированных систем представляет некоторый риск с точки зрения характеристик их функционирования, таких как надежность и безопасность.

При рассмотрении вопроса о том, достигнута ли функциональная безопасность, важны несколько аспектов, например оперативность системы ИИ и наличие или отсутствие «супервизора». В данном случае «супервизор» служит для валидации и одобрения автоматических решений, принятых системой. Такой «супервизор» может быть технической управляющей функцией, хотя в некоторых случаях такая супервизорная функция не может применяться, например в случае очень сложных решений или для систем с МО, которые научились чему-то новому. Например, вторая система, оснащенная инструментами безопасности, может быть добавлена для критически важных управляющих действий, она придана функции безопасности для резервных компонентов в соответствии со стандартами функциональной безопасности, например *ГОСТ Р МЭК 61508-1*.

Другой способ использования «супервизора» — вмешательство человека в критической ситуации или подтверждение человеком решений системы. Один из способов — добавить систему (уровень использования C или D — по 6.2), чтобы помочь человеку-супервизору заметить возможные последствия принятия решения. Примером может быть система моделирования, которая дает информацию по принципу «что, если…» для разных решений и может проверить последствия. Но даже если человек включен в контур и наблюдает за действиями системы, иногда это не может снизить риски до приемлемого уровня, а может даже привести к дополнительным рискам.

Пример — Водитель не замечает гололеда на дороге и может взять на себя управление автомобилем, не понимая, почему автопилот ведет машину так осторожно.

Адаптивность системы ИИ также является фактором, особенно с точки зрения изменения поведения системы во времени, как это происходит в системах на МО. Такие системы могут адаптироваться к изменению окружающей среды (например, через петлю обратной связи или с помощью функции оценки качества) и могут постепенно демонстрировать даже совершенно новые функции. Недостаток таких обучаемых систем состоит в том, что они могут отходить от изначальных спецификаций и поэтому трудно поддаваться валидации. По этой причине следует очень тщательно взвешивать решение о применении таких систем на более высоких уровнях использования A—C — по 6.2.

В таблице 3 (*[2]*, таблица 1) приведены термины и понятия, связанные с уровнем автоматизации.

Таблица 3 — Соотношение автономии, гетерономии и автоматизации (*[2]*, таблица 1)

Объект	Вид	Уровень автоматизации	Описание
Автоматизированная система	Автономная	Автономность	Система способна модифицировать сферу своей работы или свои цели без вмешательства извне, без контроля и надзора
	Гетерономная	Полная автоматизация	Система может выполнить поставленную задачу до конца без вмешательства извне
		Высокая автоматизация	Система частично выполняет поставленные задачи без вмешательства извне
		Автоматизация обусловленная	Выполнение конкретных задач с высокой надежностью при наличии внешнего агента, готового взять на себя управление при необходимости
		Частичная автоматизация	Некоторые подфункции системы полностью автоматизированы, но система в целом находится под управлением внешнего агента
		Ассистирование	Система является ассистентом оператора
		Отсутствие автоматизации	Оператор полностью управляет системой

Примечание — Разделение на уровни относится к функциям автоматизации управления в любых реализациях автоматизированной системы ИИ, учитываются функции компонентов этой системы (например, бортовое оборудование, напольное оборудование и оборудование диспетчерской в системах управления на железной дороге).

8.3 Степень понятности и объяснимости

Часто аспекты понятности и объяснимости объединяются под термином «понятность». Но эти термины следует различать.

В *[2]*, 5.15.6, объяснимость определяется как свойство системы ИИ отражать важные показатели, влияющие на результаты системы ИИ, в виде, понятном для людей, при этом понятность по *ГОСТ Р 59276—2020 (статья 3.10)* определяется как свойство системы ИИ, заключающееся в возможности открытого, исчерпывающего, доступного, четкого и понятного представления информации заинтересованным лицам о внутренних процессах в системе ИИ.

В частности, информация о модели, лежащей в основе процесса принятия решений, вероятно, будет актуальной. Использование систем с низкой понятностью может представлять риск с точки зрения несмещенности, защищенности и подконтрольности решений. Более того, оценка качества таких систем может быть затруднена. С другой стороны, высокая степень понятности может приводить к некоторой путанице из-за информационной перегрузки или может приводить к конфликтным ситуациям в сфере защиты личных данных, защищенности системы, конфиденциальности или охраны прав интеллектуальной собственности. Желаемый уровень объяснимости может часто быть достигнут без высокого уровня понятности. Поэтому важно найти надлежащий уровень понятности, чтобы разработчики могли идентифицировать и исправлять ошибки в системе, обеспечив при этом доверие к системе со стороны пользователя.

При программировании без использования ИИ намерения и знания программиста обычно закодированы с помощью процесса логического вывода, и можно проследить через код, как и почему определенное решение было принято программой. Это можно сделать либо через обратное отслеживание и отладку программы, либо путем обратной инженерии. В отличие от таких систем решения, принимаемые моделями ИИ, особенно сверхсложными или полученными с помощью алгоритмов МО, оказываются гораздо менее понятными для человека. Способы кодирования знаний в структуре модели и способы принятия решений часто совершенно отличаются от того, как человек рассуждает в процессе принятия решения.

Высокий уровень объяснимости предотвращает непредсказуемое поведение системы, но иногда ему сопутствует низкий уровень функционирования системы в целом с точки зрения качества решений из-за ограничений существующей технологии объяснимости (она ограничивает объем информации в модели, чтобы иметь возможность интерпретации вывода). В данном случае выбор часто приходится делать между объяснимостью и функционированием системы в целом. Кроме того, важным фактором может быть актуальность информации о процессе принятия решений в системе ИИ. Может оказаться, что система выдает ясную и понятную информацию о своем процессе принятия решений, но эта информация неточная и неполная.

Объяснимый ИИ может также использоваться как помощник при анализе данных после события, потребовавшего вмешательства оператора, когда данные на входе, иногда транзитные, записываются и воспроизводятся.

Следовательно, желательно включать оценку понятности и объяснимости в общую оценку системы ИИ. При этом следует принимать во внимание следующее:

- достаточно ли доступна информация о системе;
- понятна ли система или выдает ли она информацию в понятном виде (возможно, косвенным образом) заданному получателю (получатель объяснений может быть разным, в зависимости от контекста: разработчик системы, первые пользователи системы или, в некоторых случаях, посторонние люди);
- выдает ли система правильные, полные и воспроизводимые результаты без сбоев;
- является ли полученное от системы адекватным требуемой степени доверия.

Существуют несколько концепций и стратегий оценки понятности и объяснимости, например приведенные в *[11]*. Кроме того, эмпирическая оценка процесса принятия решений для сложных моделей может происходить, например, путем анализа сверточной нейрононной сети через визуализацию компонентов ее внутренних слоев. Цель — сделать процесс принятия решения в сети более объяснимым

путем наблюдения за тем, как особенности данных на входе влияют на полученный моделью вывод. Анализ информации на выходе СНС может выполняться путем инспектирования ее внутреннего состояния экспертом. Даже если доступ к данным о внутреннем состоянии модели невозможен, подходы, подобные рандомизированной выборке входных значений для (randomized input sampling for explanation, RISE), могут дать некоторое понимание для некоторых видов сетей.

Даже системы, которые традиционно считаются удовлетворительно объяснимыми путем визуального анализа (например, дерево решений), могут быстро стать настолько сложными, что не поддаются пониманию, если эти системы применяются в реальном мире. В ситуациях, когда нужно получить результат, поддающийся интерпретации, можно применить такие инструменты, как деревья оптимальной классификации или методы дистилляции ансамблей деревьев для снижения сложности и создания условий для инспекции экспертом.

Вообще говоря, даже если полностью объяснимый ИИ недостижим в настоящее время (подобный классу II технологии ИИ), методичное задокументированное оценивание возможности интерпретации модели может быть применено с точки зрения риска при тщательной оценке последствий для угрозы функциональной безопасности в случае ненадлежащих решений. Это помогает сравнить и выбрать модель и может прояснить ситуацию при анализе отказа после события.

8.4 Проблемы, связанные с окружающей средой

8.4.1 Сложность внешней среды и неопределенность спецификаций систем

Системы ИИ часто используются в такой среде, сложность которой не позволяет человеку провести ее анализ и описать ее. Технология ИИ может автоматически генерировать правила или применять суждения, не полагаясь на созданные человеком представления аналитических, подробных и сложных условий окружающей среды. Далее жизненный цикл разработки систем ИИ может начинаться при неопределенных спецификациях и неопределенных целях. Такие спецификации могут быть заданы в форме функциональных или нефункциональных требований.

Неопределенность спецификаций может привести к трудностям при утверждении функциональных характеристик, связанных с безопасностью. Сложность среды только ухудшает ситуацию. Даже определение допустимого уровня функциональной безопасности может быть невозможным из-за неточных спецификаций, поскольку определение функции безопасности зависит от заданной спецификации.

Для приложений функциональной безопасности важно пояснение функциональной полноты (см. *ГОСТ Р ИСО/МЭК 25010*). Вопросы функциональной полноты могут решаться с помощью подробной спецификации или при полном покрытии всей сложной среды (например, данными для обучения), или сочетая эти два подхода. Рекомендации приведены в 9.3.

Еще одна особенность сложности систем ИИ заключается в том, что, хотя их модели часто детерминированы, их выходные данные могут казаться вероятностными. Например, если взять очень сложную среду, которая может быть представлена большим пространством состояний и при этом среда постоянно изменяется и расширяется, может быть сложно гарантировать, что модель, которая хорошо обобщает поведение при конечном числе состояний, сможет реагировать надлежащим образом на каждое возможное состояние среды.

Эффект функционирования в сложной и не до конца определенной среде может привести к новому типу неопределенности, который находится за пределами существующих оценок функциональной безопасности.

Степень демонстрации адекватности моделей для целевого приложения является важным фактором. Кроме того, важно, что неопределенность модели из-за возможности некорректных прогнозов или неверных вычислений рассматривается с точки зрения планирования поведения и функционала.

Чтобы применить стохастическую концепцию к операционной среде, расширение стохастических допущений, обычно используемых в области функциональной безопасности для случайных отказов оборудования и «проверенных на практике» программных средств, является относительно новым подходом, который также может быть применим к технологиям ИИ, см. 10.2.5.

8.4.2 Проблемы, связанные с изменением среды

8.4.2.1 Дрейф данных

Дрейф данных — явление, связанное с тем, что распределение входных данных в бизнес-процессе отличается от использованного при обучении, что приводит к ухудшению функционирования модели, включая и безопасность. Дрейф данных часто связан с неполным представлением области допустимых значений входных данных во время обучения. Это явление может объясняться отсутствием учета се-

зонных изменений в данных на входе, непредвиденным вводом данных операторами или добавлением новых сенсорных устройств, которые появились в качестве источника вводимой информации.

Компоненты, включающие технологию ИИ, могут инспектироваться для обнаружения источников дрейфа данных в контексте анализа рисков и надлежащие меры могут планироваться в случае необходимости.

Некоторые случаи дрейфа данных могут быть отнесены к неверному выбору данных при создании модели. Типичными примерами являются выбор неподходящих данных при обучении модели, таких, где распределение данных на обучении не отражает истинное, наблюдаемое в контексте применения распределение данных, или пропуск важных примеров в данных для обучения. Такие проблемы могут исправляться с помощью улучшения данных на входе и переобучения.

Дрейф данных может быть вызван внешними факторами, например сезонными изменениями или изменениями в процессе, который способствует дрейфу. Примеры включают замену сенсора новым, который имеет новое напряжение смещения, или сенсор воспринимает разницу в условиях освещения при обучении и в данных, которые раньше не получал. Может быть целесообразно дать модели возможность справиться с дрейфом данных уже после ее развертывания, когда дообучение недоступно. В таких случаях модель создается таким образом, чтобы она оценивала факторы коррекции исходя из особенностей данных на входе или позволяла коррекцию супервизором.

Устройство модели должно позволять получение безопасных данных на выходе даже при вводе на вход никогда не вводимых ранее данных. Важно понимать, что использование надлежащих практик создания модели, например создание достаточно разнообразного набора обучающих данных, не снижает важности тщательного анализа для ответа на вопрос, может ли полученная в результате модель быть обобщена на данные в промышленной среде. Кроме того, если модель выдает небезопасные данные на выходе, важно провести тщательный анализ причин получения небезопасных данных на выходе и определить способ выхода системы из опасного состояния.

8.4.2.2 Изменение закономерности

Изменение закономерности означает изменение соотношения между переменными на входе и данными на выходе модели, она может сопровождаться изменениями в распределении вводимых данных. Например, данные на выходе могут использоваться для отслеживания приемлемого минимального расстояния оператора исходя из данных расстояния, полученных сенсором времени полета (данные на входе). Если принятые пороги безопасности изменяются под влиянием внешних факторов (например, возросшая скорость машины, не учтенная в модели), то произойдет изменение закономерностей, хотя и процесс, и данные на входе остались неизменными.

Системы в идеале имеют встроенные формы обнаружения дрейфа, отличая дрейф от шума в системе и адаптируясь к изменениям со временем. Потенциальные подходы включают модели, подобные методу раннего обнаружения дрейфа (early drift detection method, EDDM), обнаруживающие дрейф методом опорных векторов или отслеживающие ошибки логического вывода в процессе обучения с поправкой на обнаружение дрейфа и потенциальную адаптацию. Обнаружение дрейфа подразумевает некоторую форму мониторинга рабочего цикла и обновления модели, которые могут вносить рекомендуемые изменения в структуру и безопасность системы на уровне ПО или системы в целом (например, знать, когда функционально безопасно можно провести обновление, отслеживать неудавшиеся обновления).

Обработка изменений закономерностей часто происходит путем отбора подмножеств имеющихся обучающих данных или путем присвоения весовых коэффициентов отдельным обучающим объектам и затем переобучения модели.

8.4.3 Проблемы обучения на данных из внешней среды

8.4.3.1 Алгоритмы избыточной подстройки под вознаграждение

Алгоритмами избыточной подстройки под вознаграждение называют методы, когда ИИ находит способ «обыграть» свою функцию вознаграждения и найти «оптимальное» решение поставленной задачи. Это решение, хотя и более оптимальное в математическом смысле, нарушает допущения и ограничения в заданном сценарии в реальном мире. Например, система ИИ обнаруживает людей с помощью камеры и может решить, что получит большое вознаграждение в процессе обучения, если будет постоянно обнаруживать людей и отслеживать их с помощью своих сенсоров, при этом, возможно, не замечая критически важных событий на других участках. Такая тенденция может быть пресечена с помощью функций состязательного вознаграждения через независимую систему, которая может верифицировать полученные значения вознаграждения со стороны первичной функции, использующей технологии ИИ, и впоследствии обучится и адаптируется, чтобы противостоять первичной системе. Другая

опция — предварительно обучить несвязанную функцию вознаграждения на основе только желаемого результата, без прямой связи с первичной функцией.

8.4.3.2 Безопасное исследование

Безопасное исследование представляет собой задачу, когда требования безопасности должны быть принудительно установлены во время сбора данных и обучения, что ограничивает разнообразие данных, которые получала модель на этапе обучения. Проблема безопасного исследования особенно остра, если агент имеет возможность исследовать среду и изменять ее. Это является проблемой не только, например, для сервисных роботов, беспилотных летательных аппаратов и других физических объектов, но и для программных агентов, использующих обучение с подкреплением для исследования своего рабочего пространства. В этих контекстах исследования пространства обычно подкрепляются вознаграждением, поскольку дают системе новые возможности для обучения. Очевидно, что самообучающаяся система должна следовать соответствующим протоколам функциональной безопасности в процессе исследования. Система, которая управляет параметрами бизнес-процесса и использует функцию случайного исследования среды, при этом не будучи отсоединена от самого бизнес-процесса, может представлять большую опасность.

8.5 Устойчивость к состязательным вводам данных и злонамеренному воздействию

8.5.1 Введение

При оценке надежности системы ИИ, важно определить целостность реакции функциональной безопасности на вредоносные атаки и злонамеренные воздействия.

В целом можно выделить в сфере ИИ два типа воздействия, направленного на возможный сбой в системе. Первый тип — те входящие данные, которые разрушают целостность выполнения программы (например, переполнение буфера или целочисленное переполнение), второй тип — те, которые заставляют модели ИИ вычислять неверный вывод без сбоев на уровне программы. Для решения первого типа проблем следует рассмотреть традиционные для ИТ требования безопасности, установленные в *ГОСТ Р ИСО/МЭК 18045, ГОСТ Р ИСО/МЭК ТО 19791, ГОСТ Р ИСО/МЭК 27001, ГОСТ Р МЭК 62443-2-1* и *ГОСТ Р МЭК 62443-3-3*. Приведенные стандарты описывают процессы аудита и сертификации единых требований к безопасности в сфере ИТ, которые могут применяться и к системам ИИ и не обсуждаются далее в настоящем стандарте. Однако для решения второго типа проблем недостаточно использования лучших практических подходов и следования нормам стандартов для систем, не использующих ИИ. В 8.5.3 приведено описание второго типа проблем, влияющих на работу системы, с примерами состязательности естественного происхождения.

Примечания

1 Настоящий стандарт ограничивается достижением функциональной безопасности даже в присутствии направленной на ИИ угрозы безопасности. В нем не описывается, каким образом злонамеренное действие, возникающее по причине угрозы кибербезопасности, может быть пресечено. Меры, направленные на обеспечение целостности функциональной безопасности, важны, если разумно предсказуемая киберугроза может повлиять на функциональную безопасность.

2 Характеристики, обеспечивающие отсутствие злонамеренных вводных данных, могут противоречить тем, которые гарантируют характеристики функциональной безопасности.

3 Характеристики, обеспечивающие устойчивость к состязательным атакам, могут противоречить тем, которые гарантируют характеристики функциональной безопасности. Это рассматривается как фактор высшего уровня системной пригодности.

8.5.2 Общие защитные меры

В соответствии с надлежащими предосторожностями в сфере функциональной безопасности в качестве первого шага, обеспечивающего функционально безопасную работу, предлагается применение функций контроля, которые берут на себя управление системой в случае обнаружения проблемы в области функциональной безопасности, таким образом обеспечивается отсутствие вреда со стороны системы ИИ.

Для систем, требующих высокого уровня функциональной безопасности, требуется повышенное внимание как к случайным отказам, так и к систематическим ошибкам. В целом отказы и ошибки устраняются в соответствии с лучшими практиками (например, путем повышения защиты, робастности, тестирования и верификации). Кроме того, специальные меры противодействия в области МО важны для дальнейшего снижения рисков, связанных с дополнительными видами отказов и ошибок, свойственных технологии ИИ.

8.5.3 Атаки на модель ИИ: состязательное МО

Модели систем ИИ, особенно наиболее сложные из них, например нейронные сети, могут иметь особые слабые места, несвойственные другим типам систем. Необходимо рассмотреть их дополнительно, особенно в контексте функциональной безопасности. Примеры специфичных для моделей проблем включают состязательное МО и др.

Состязательное МО — это один из типов атаки на системы ИИ, который привлекает большой интерес в последнее время. Часто становится возможно обмануть модель ИИ и заставить ее выдавать абсолютно разные результаты путем небольших возмущений на входе, которые создаются с помощью процесса оптимизации. Если вводятся изображения, эти возмущения незаметны в целом для человека и могут хорошо маскироваться и в цифровых данных на входе. Хотя эти возмущения обычно не случайные, нельзя исключать, что аппаратные отказы и системный шум, уже присутствующие во входных данных, вызывают существенный сдвиг в данных на выходе. Интересно то, что состязательные примеры обычно хорошо воспроизводятся в моделях разной архитектуры и ее внутренних компонентах. Этот факт, при большом количестве известных архитектур моделей и заранее обученных моделей в так называемых «зоопарках», делает возможным практическое применение состязательных примеров, что представляет серьезную угрозу системам, использующим ИИ.

Даже система, которая кажется устойчивой к модификациям информации на входе (например, такая, которая использует локальную, не взятую из облака модель ИИ, напрямую соединенную с сенсорами), не застрахована от какой-либо состязательной атаки. Например, могут быть осуществлены физические атаки на модели, даже те, которые считаются «черным ящиком» без доступа к составляющим внутренней модели. Также могут быть введены состязательные примеры в процесс логического вывода модели, создавая упомянутые возмущения за счет обычных стикеров, приклеенных к предметам, и заставляя результирующий вывод отклоняться от оптимального решения.

Когда устройство входа модели ИИ неустойчиво против состязательных атак, чистый эффект таких атак может быть точно оценен до того, как придется решать, следует ли применять меры противодействия и какие именно и какое их количество считать достаточным. Возможность состязательных атак в реальной системе варьируется в широких пределах, в зависимости от того, как внедрена модель ИИ. Например, эта возможность очень зависит от систем, окружающих технологию ИИ, включая получение входных данных через сенсор (например, камеру) и предварительную обработку данных. Более того, анализ возможных агрессоров и их жертв тоже важен, если единственные возможные жертвы совпадают с возможными агрессорами, в некоторых случаях можно просто не применять защитных мер.

Одна из предлагаемых мер для решения проблемы называется состязательным обучением. По сути, состязательное обучение пытается обучить систему ИИ на примерах, пытается заставить модель закодировать знания об ожидаемых данных на выходе при такой атаке. Другое направление — попытка устранить искусственно введенное возмущение.

Типы моделей, подверженные состязательным атакам, в целом являются робастными по отношению к шуму. Чтобы модифицировать ввод и повысить робастность по отношению к злонамеренному, направленному шуму, применяют различные схемы рандомизации. Методы включают случайное изменение размера и замощение изображения, случайные самоансамбли (random self-ensembles) и разные трансформации данных на входе, такие как сжатие JPEG или модификация битовой глубины на изображениях. Хотя такие методы эффективны, но не всегда достаточны. В свою очередь, если трансформации на входе используются как слой защиты от состязательных примеров, эффективность упомянутых мер можно оценить, используя алгоритм ожидания преобразования (expectation over transformation, EOT).

Использование моделей на основе нелинейных компонентов делает их менее подверженными состязательным атакам за счет возрастающих вычислительных ресурсов. Ансамбли моделей часто используются для создания более робастной общей модели путем диверсификации. Однако диверсификация не всегда в должной степени защищает систему от состязательной атаки.

В дополнение к атакам, изменяющим данные на выходе в функционирующей системе, можно ввести возмущение во время обучения модели, введя злонамеренные данные на этапе обучения, что называется «отравление модели». Учет необходимости защиты процесса обучения на этапе сбора данных очень важен.

8.6 Проблемы аппаратных средств, связанные с применением ИИ

Технология ИИ не может сама по себе принимать решения, она полагается на алгоритмы, программные средства, применяющие алгоритмы, и аппаратные средства, выполняющие эти алгоритмы. Сбои в оборудовании могут нарушить правильное выполнение алгоритма, нарушив ход исполнения (вызывая ошибки доступа к памяти, помехи в данных на входе, например от сенсоров), повреждая данные на выходе и в целом вызывая ошибочные результаты. В настоящем подразделе описываются некоторые аспекты аппаратных средств при использовании технологии ИИ, которые могут повлиять на функциональную безопасность. Коротко, надежное оборудование так же важно для систем ИИ, как и для систем без ИИ. Так же, как оборудование, используемое для выполнения обычных программ, оборудование для выполнения программ технологии ИИ может быть подвержено случайным отказам. Перечень отказов приведен в *ГОСТ Р МЭК 61508-2* и *[9]*.

8.7 Зрелость технологии

Зрелость технологии отражает то, насколько зрелой и свободной от ошибок является конкретная технология в конкретном контексте применения. Менее зрелые и новые технологии, используемые при разработке систем ИИ, могут привнести риски, которые ранее не были известны и которые трудно оценить. Для зрелой технологии существует разнообразный опыт применения, что позволяет легче выявить риски, оценить их и принять против них меры. Однако зрелые технологии приводят к опасности сниженного осознания их потенциального воздействия на риск со временем, так что положительные эффекты зависят от постоянного мониторинга риска (например, на основе сбора полевых данных), а также в части соответствующего обучения сотрудников и от технического обслуживания системы.

9 Приемы верификации и валидации

9.1 Введение

В настоящем разделе приведено описание различий между методами верификации и валидации для систем с ИИ по сравнению с системами без ИИ, а также описание некоторых соображений для решения или смягчения проблем, возникающих из этих различий, применимых к функциональной безопасности. В настоящем разделе рассматриваются четыре важных аспекта указанных различий, хотя потенциальные различия не ограничиваются теми, которые здесь описаны.

Настоящий раздел посвящен моделям, которые созданы в процессе МО. В 8.1.2 этот класс моделей рассмотрен в качестве главной проблемы обеспечения функциональной безопасности системы ИИ. Это происходит потому, что функциональная безопасность других типов технологии ИИ может быть иногда достигнута путем применения принципов из существующих стандартов функциональной безопасности, которые описаны в разделе 7. Содержание настоящего раздела предназначено в основном для применения на уровнях использования от А1 до С класса II систем ИИ (см. таблицу 1, 6.2).

Для создания функционально безопасных систем, использующих ИИ на основе информационно обоснованных моделей, принимается во внимание то, что технология ИИ не создается правилами, как это происходит в системах, созданных без применения ИИ. Это значит следующее:

- то, чего нет в данных, выучить нельзя;
- то, что есть в данных, может быть выучено, но не всегда в совершенстве.

Более того, просто иметь данные в большинстве случаев оказывается недостаточно. Разметка имеет решающее значение при применении обучения с учителем. Неправильная разметка является одной из главных причин ошибок в процессе обучения. Подробно описанный процесс подготовки данных важен при обращении к этим аспектам.

Если модель получена на основе данных, содержание настоящего раздела может применяться для обучения и валидации наборов данных.

Примечание — Термины «валидация» и «верификация» могут относиться к разным понятиям в разных сферах. В контексте МО валидация означает шаг в процессе для проверки сходимости разрабатываемой модели для завершения процесса обучения ИИ, что совершенно отличается от понятия верификации и валидации в сообществе экспертов по функциональной безопасности. Сходимость модели является важным предварительным условием для тестирования, но она не означает гарантий качества конечного продукта. Например, проблема незаслуженного вознаграждения возникает из-за модели, которую субъективно создали для максимизации заданной

функции вознаграждения. В настоящем стандарте термины валидация и верификация почти исключительно используются в контексте функциональной безопасности.

9.2 Проблемы верификации и валидации

9.2.1 Отсутствие априорной спецификации

На этапе обучения моделей МО выбор обучающих данных (вместе с определением функции потерь, если применимо) заменяет определение формализованной спецификации операционного поведения. Это приводит к проблемам с прослеживаемостью отдельных аспектов поведения ввиду отсутствия отдельных утверждений спецификации. Вместо этого информация, которая заменяет дискретные операторы спецификации, неявно содержится в обучающем наборе данных. Хотя преимуществом МО является его способность извлекать или приобретать знания из плохо структурированных данных, отсутствие заранее определенной спецификации может вызвать серьезную проблему верификации и валидации, а также оценивания неопределенности.

Другим источником риска является наличие систематической ошибки или неполноты данных, используемых для обучения модели. Необходимо применение техник для проверки обоих источников риска.

9.2.2 Неотделимость определенного поведения системы

При разработке ПО для приложений без ИИ, связанных с функциональной безопасностью, все риски, которые были описаны в разделе 8, и в процессе анализа опасностей и оценки рисков (hazard and risk analysis, HARA) были идентифицированы как «допустимые», сопоставлялись с одной или несколькими мерами снижения рисков. Необходимо должным образом реализовать внедрение мер снижения рисков и объяснить их роль для функциональной безопасности. Также важно, чтобы меры не создавали помехи друг другу для проведения верификации, валидации и оценки каждой из них.

С другой стороны, многие технологии ИИ можно считать «черным ящиком», поскольку их внутреннее поведение и основы процессов принятия решений трудно понять человеку. Это означает, что, если обучающий набор данных содержит некоторые данные, предназначенные для снижения определенного риска, их влияние на обученную модель не может быть определено или протестировано отдельно для каждого риска. Кроме того, если для дополнительного снижения риска будут добавлены некоторые дополнительные обучающие данные, эти данные могут повлиять на существующие меры снижения других рисков. Это усложняет верификацию и валидацию моделей МО.

9.2.3 Ограничение покрытия тестами

По сравнению с тестированием ПО без ИИ тестирование технологий с ИИ гораздо сложнее. Обычно для ПО разрабатываются и выполняются два типа тестов: один из них связан со структурой описания проблемы, а другой — со структурой ПО. В ПО без ИИ эти два теста в определенной мере соответствуют друг другу, что обеспечивает эффективность тестирования. К сожалению, большинство технологий ИИ не обладают этим свойством, что не позволяет применять существующие техники для ПО без ИИ (включая описанные в существующих стандартах функциональной безопасности). Это различие заслуживает особого внимания при разработке тестов для технологий ИИ (в особенности тех, которые основаны на МО).

9.2.4 Непредсказуемость

Как отмечено в 8.4.1, выходные данные системы ИИ часто называют непредсказуемыми или вероятностными по своей природе, хотя сам алгоритм может быть детерминированным. Смягчить последствия можно путем систематического проведения верификации и валидации с тщательным учетом особенностей системы ИИ. Технология «объяснимого» ИИ может появиться в будущем, но пока в приоритете остаются решения, максимально ориентированные на максимально точное моделирование конкретных процессов.

Кроме того, явный непредсказуемый или вероятностный характер технологии ИИ, а также другие причины, включая те, что были рассмотрены в 9.2.3, снижают эффективность или применимость методов тестирования, не связанных с технологиями ИИ, таких как тестирование на основе «белого ящика». См. *[12]*, раздел 9, для получения информации об альтернативных решениях для тестирования на основе «белого ящика», применимых для систем ИИ.

9.2.5 Дрейфы и долгосрочные меры по снижению рисков

Дрейфы (см. 8.4.2 и 8.4.3) являются еще одной причиной неопределенности при установке системы в долгосрочной перспективе. Даже при проведении систематического и всестороннего анализа их поведения в рабочей среде модели ИИ не могут быть полностью избавлены от дрейфа закономерно-

стей в данных и дрейфа самих данных, поскольку набор обучающих данных характеризуется предвзятостью условий времени обучения. Чтобы преодолеть такой дрейф, предлагается несколько методов переобучения и обновления модели для большинства реальных вариантов применения систем ИИ.

Кроме того, обновление ПО является важной задачей, особенно в случаях, связанных с функциональной безопасностью. Важно принимать во внимание существующие оценки и подходы, связанные с процессами обновления, на самых ранних этапах проектирования системы.

9.3 Возможные решения

9.3.1 Общие положения

9.3.1.1 Подходы к снижению рисков

В целом существует по крайней мере два подхода к реализации надежной верификации и валидации решений, созданных на основе моделей с данными.

Один подход, вероятно, более сложный, заключается в анализе сгенерированной модели для извлечения понятной человеку информации об ожидаемом поведении модели. Теоретически, если поведение становится полностью объяснимым для человека, мы можем рассматривать технологии и системы ИИ как ИИ класса I. Этот подход подробнее описан в 9.4.

Другой подход предполагает косвенную оценку достигнутых уровней снижения рисков безопасности путем анализа процесса создания системы ИИ в процессе разработки. Хотя тестирование ИИ на основе МО не всегда можно завершить, дополнительный анализ и проверка процессов разработки и входных данных могут позволить систематически снижать риски нежелательного поведения. В 9.3 главным образом описан именно этот подход.

9.3.1.2 Показатели ИИ, верификация и валидация безопасности

Как правило, такие показатели, как точность, используются при обучении алгоритмов МО. Эти показатели необходимы для управления процессом обучения ИИ. Хотя более высокая точность предполагает лучшее качество реализации функциональной безопасности, этого бывает недостаточно для обеспечения требуемых свойств функциональной безопасности. Важно, чтобы меры по снижению рисков, описанные в настоящем разделе, применялись параллельно или последовательно с оценкой используемых показателей ИИ, особенно на этапах разработки, подготовки и тестирования данных.

9.3.2 Связь между распределением данных, анализом опасностей и оценкой риска

Для процессов, управляемых данными, критически важна взаимосвязь между рисками, выявленными в результате анализа опасностей и оценки риска, и распределением данных. В этом случае возникает вопрос о достаточности обучающих и тестовых данных, получаемых системой ИИ, для выработки определенного поведения. Результаты анализа опасностей и оценки рисков и используемые наборы данных должны соответствовать. Этот подход можно считать аналогичным подходу, реализуемому для ПО без ИИ, в рамках которого были определены меры по снижению риска.

Независимо от того, является ли начальная спецификация предопределенной или она получена из примеров данных, первоочередной задачей является максимально точное определение и ограничение предметной области системы. Границей может быть либо интервал входных данных, либо характер реального применения. Важно на ранних стадиях установить соответствие показателей проверки набора данных и действий по верификации и валидации соответствующей предметной области.

Также кроме сбора и изучения набора данных необходимо выполнить логический анализ распределения входных данных. Он относится к результатам анализа опасностей и оценки риска, так что точки распределения данных в наборе данных идентифицируются как соответствующие каждому выявленному риску (например, см. *[13]*, где установлено, что качество данных является ключевым требованием для технологий ИИ).

Кроме того, даже если входной набор данных хорошо разработан, невозможно гарантировать, что процесс обучения позволит закодировать предполагаемое поведение выходной модели в соответствии со всеми выявленными в процессе наблюдений за распределением данных рисками. При обучении могут возникать как систематические, так и случайные ошибки, которые могут привести к нарушению функциональной безопасности. Хотя обнаружение таких сбоев в максимально возможной степени является одной из целей тестирования на этапе верификации и валидации, сокращение ошибок обучения также важно на этапе обучения.

9.3.3 Подготовка данных, верификация и валидация модели

Принимая во внимание, что цель разработки наборов данных для обучения и тестирования определяется в соответствии с результатами анализа опасностей и оценки риска, следующим шагом должно

стать обеспечение соответствия этих наборов данных определенным целям. Выполнение требования к данным включает четыре важных критерия:

a) все ли сценарии, относящиеся к функциональной безопасности, выявленные в процессе анализа опасностей и оценки риска, включают соответствующие данные в наборы данных;

b) охватывают ли тестовые данные все возможные варианты ситуаций, провоцирующих такой риск для каждого риска, выявленного в результате анализа опасностей и оценки риска;

c) являются ли тестовые данные достаточно большими и достаточно разнообразными, чтобы позволить охватить все возможные состояния системы после обучения для каждой ситуации, провоцирующей риски;

d) являются ли результаты выбранных данных тестового примера стабильными при всех возможных вариациях входных данных, которые согласно анализу могут быть классифицированы как принадлежащие к той же группе, сценарию или варианту использования для каждой ситуации, провоцирующей риски.

Предполагается, что каждое тестовое задание способно дать ответ на каждый из четырех вопросов. Ниже представлен пример ответов на вопросы, актуальные для любой технологии ИИ, испытание которой связано с четкими, правильными и ожидаемыми ответами («тестовые оракулы»). В этих примерах также учитывалась погрешность данных по *ПНСТ 839—2023*.

Для a):

- четко определяют признаки наборов данных в соответствии с рисками, выявленными в результате анализа опасностей и оценки риска.

Для b):

- проверяют наличие тестового набора данных для каждого установленного признака набора данных в соответствии с выявленными рисками;

- проверяют распределение других признаков и оценивают возможную смещенность данных для всех тестовых данных, определенных в соответствии с выявленными рисками. В этих целях можно использовать существующие технологии генерации тестовых данных для ПО без ИИ (например, комбинаторное тестирование). Также см. *[12]* и *[14]*;

- при подозрении, что набор данных содержит непреднамеренное, но тем не менее систематическое смещение, рассматривают возможность сбора дополнительных тестовых данных. В некоторых случаях решением может стать синтез тестовых данных, если на основе реальных данных невозможно получить достаточное разнообразие, представительность и охват. См. *[12]*. Разработчик также может удалить реальные данные, чтобы сбалансировать набор данных.

Для c):

- меры по снижению рисков, указанные в b), могут применяться для повышения разнородности существующих признаков;

- оценивают процессы сбора и подготовки данных, чтобы исключить возможность включения смещенных данных, см. *ПНСТ 839—2023*;

- объем тестовых данных определяется на основе входных данных, включая (среди прочего) предполагаемую вероятность снижения риска (полученную в результате анализа опасностей и оценки риска), и данных, необходимых для обучения (полученных в процессе мониторинга показателя точности для подмножества обучающих данных). Также необходимо учитывать сложность предметной области, чтобы смягчить сдвиги при распределении данных, возникающие из-за множества неконтролируемых факторов (например, времени, погоды, местоположения).

Для d):

- убеждаются, что все случаи избыточной подгонки к обучающим данным были обнаружены на этапе разработки. Для этого важно добиться независимости обучающих и тестовых данных, которая может быть обеспечена с помощью внедрения процессов управления и оценки качества разработки, использования инструментальных средств или посредством достижения независимости команд и организаций, выполняющих тестирование. См. *ГОСТ Р МЭК 61508-1—2012* (раздел 8) или *[15]*. Также случаи избыточной подгонки могут быть обнаружены при проведении перекрестной проверки, когда модели обучаются на разных обучающих данных, а производительность оценивается на основе удержанных данных;

- убеждаются, что обученные модели достаточно устойчивы, используя следующие подходы:

a) создание нескольких моделей разных размеров и использование моделей меньшего размера при условии достижения других целей (большие модели могут повышать чувствительность);

b) применение технологий, повышающих надежность (например, регуляризации или рандомизированного обучения);

c) численная и прямая оценка надежности.

- производят поиск данных, влияющих на устойчивость (например, при помощи метаморфического тестирования, аугментации данных, генеративных состязательных сетей, состязательного обучения, генерации состязательных примеров или обнаружения состязательного примера);

- убеждаются, что в обучающем и тестовом наборах данных отсутствуют преднамеренные модификации или изменения, что может повлечь проверку достоверности источников данных или процессов сбора данных;

- рассматривают возможность применения технологии объяснимости для анализа поведения выходной модели (см. 9.6).

В приложении С приведена информация о применимых процедурах и методах.

Затраты на реализацию мер по смягчению последствий могут существенно различаться в зависимости от глубины исследований и используемых уровней комбинации критериев, представленных в b), а также выбранной технологии. При планировании верификации и валидации необходимо исходить из требуемого уровня функциональной безопасности и других критериев.

9.3.4 Выбор показателей ИИ

Оценка производительности и КПЭ системы с ИИ крайне важна. В МО часто используются отдельные конкретные показатели. Хотя показатели первичны, также необходимо принимать во внимание следующие аспекты:

Примечание — Показатели, как правило, являются не единственным, а лишь одним из способов оценки безопасности системы ИИ;

- важность и достоверность показателей определяется в зависимости от объема данных, доступных для обучения, проверки и тестирования. Объем данных влияет на доверие к показателю с определенным уровнем достоверности (например, 95 %) в зависимости от количества выполненных проверок;

- показатели агрегируют большой объем информации в одно число (или несколько чисел), что может отвести внимание от вопросов, связанных с безопасностью. Для выявления недостающей информации можно использовать различные показатели, такие как, например, неправильная классификация безопасности при оценке целевых характеристик или КПЭ;

- мониторинг в процессе эксплуатации: сбор данных о производительности системы ИИ на этапе эксплуатации важен для оценки достижения производительности и КПЭ, промежуточный мониторинг важен в случаях, когда выводы о безопасности системы неверны.

Оценка робастности по *ГОСТ Р 70462.1* подразделяется на три основные категории: статистические, формальные и эмпирические тесты.

9.3.5 Тестирование на системном уровне

В комплексных системах, где ИИ используется в качестве одного из компонентов, тестирование на системном уровне является важным дополнением к верификации и валидации на детальном уровне. Некоторые из критериев, описанных в 9.3.3, например критерии b), также применимы для тестирования на уровне системы. Тестирование на системном уровне может быть основано либо на данных, либо на сценарии (например, запуск тестового автомобиля на полигоне с моделируемыми рисками). Тестирование на системном уровне можно проводить на моделях, цифровых двойниках или в реальных условиях. Тестирование в реальных условиях весьма затратно и не всегда возможно (отчасти из-за рисков для безопасности), но оно позволяет проверить КПЭ и выявить неопознанные опасные неизвестные факторы для предотвращения неполных анализа опасностей и оценки рисков. Тестирование на моделях позволяет изучить множество сценариев как программного, так и аппаратного обеспечения. Качество и реалистичность моделей играют важную роль для достижения хороших результатов верификации и валидации. Подробнее см. 9.4.2 и 9.4.3.

9.3.6 Методы снижения рисков, связанных с ограниченными по размеру выборками данных

Подготовка достаточно больших тестовых оракулов для проверки всех результатов невозможна в рамках жизненного цикла разработки. Сравнительное тестирование, описанное в *[12]*, может использоваться для трактовки тестовых оракулов в соответствии с ожидаемыми ответами. Также необходимо оценить степень независимости различных версий тестируемой системы. Сравнительное тестирование технологий ИИ, генерируемых одним источником обучающих данных, вероятно, не будет соответствовать критериям a) и b).

Другим вариантом применения тестового оракула для выполнения всего спектра операций может быть использование модели в качестве генератора тестовых данных.

Специалистам бывает сложно создать надежный тестовый оракул для некоторых систем ИИ (например, систем ИИ, созданных на противостоянии «ИИ против ИИ», а также для обучения с подкреплением и генеративно-состязательной сети). Общие условия тестирования в этих случаях аналогичны, однако могут применяться дополнительные критерии для проверки надежности тестов. Например, качественно проверенные альтернативные разработки могут использоваться при сравнительном тестировании. Также можно вносить изменения при разработке для исключения влияния любых рисков на технологию ИИ, управляемую моделями уровня использования С, как описано в 6.2.

9.3.7 Примечания и дополнительные источники

См. *[12]*, в котором описаны альтернативные решения для тестирования на основе «белого ящика», применимые для систем ИИ.

9.4 Виртуальное и физическое тестирование

9.4.1 Общие положения

Подходы к функциональной безопасности технологии ИИ, как правило, сосредоточены на элементах системы ИИ, которые обеспечивают такие атрибуты функциональной безопасности, как, например, мониторинг, основанный на правилах, способный блокировать основную систему управления для предотвращения небезопасных действий. Эффективным и объективным способом демонстрации производительности системы является виртуальное тестирование или моделирование, когда в ходе квалификационных и сертификационных мероприятий можно проверить набор тщательно отобранных сценариев для проведения стресс-тестирования. Возможно тестирование как отдельных компонентов, так и нескольких компонентов на системном уровне. В рамках этого подхода при построении сценариев для тестирования можно использовать ограниченный выбор значений параметров сценария, сценарии на основе распределения параметров или выборки важности (см. *[4]*, С.5).

Физические тесты также важны для сопоставления результатов моделирования, проверки КПЭ и выявления неизвестных. Физические тесты гораздо более ограничены, чем моделирование, в их способности исследовать область значений из-за ограничений по стоимости и времени, но они позволяют проверить некоторые показатели, которые довольно сложно смоделировать, как, например, влияние задержек, вызванных аппаратными средствами, на петли обратной связи и каскадные эффекты. Возможно проведение структурированных тестов, в которых применяются настройки для известных сценариев, как, например, в случае тестирования интеллектуальных транспортных систем на тестовом полигоне.

9.4.2 Рекомендации по проведению виртуального тестирования

Симуляционное моделирование в целях тестирования уже давно является неотъемлемой частью функциональной безопасности. Оправданны такие устоявшиеся методы, как симуляция синхронизации и внесение неисправности, которые благотворно влияют на системы ИИ. Для сложных многомерных моделей, используемых во многих системах ИИ (таких, как нейронные сети для задач распознавания или принятия решений), симуляционное моделирование имеет ряд дополнительных преимуществ:

- в ряде случаев симуляция обеспечивает более полный охват тестами, чем при тестировании в реальных условиях. Примерами могут быть сценарии, в которых реальное тестирование может быть опасным или излишне дорогостоящим в связи с большим масштабом на возможном пространстве вывода. Для моделей большой размерности моделирование может быть использовано для перебора входного пространства и получения коррелированных результатов способами, недоступными при традиционном тестировании;

- симуляция позволяет значительно ускорить время разработки и предоставляет более широкий доступ к продуктам и обновлениям функциональной безопасности. Недавно обнаруженные опасности могут быть учтены при разработке решений для функциональной безопасности, что позволяет сократить время выполнения работ. Для сложных сред такие факторы, как сокращение задержек при разработке и появление обновлений, могут иметь решающее значение;

- симуляция позволяет использовать несколько входов для внесения неисправностей. Неисправности могут возникать на уровне системы, компонента или подкомпонента, а также в комбинациях, которые могут быть недоступны при реальных испытаниях;

- симуляция позволяет добиться эталонных данных и полностью исключить риск потенциальных систематических ошибок, вызванных реальными измерениями и настройками;

- среду симуляции легче контролировать, что позволит отслеживать все метаданные в условиях конкретного теста, то есть предотвратить любую систематическую погрешность, возникающую в реальном тесте, или потерю соответствующих метаданных.

Необходимо принимать во внимание следующие аспекты при проведении симуляционного моделирования или, в целом, виртуального тестирования для верификации и валидации технологии ИИ в системах функциональной безопасности:

- точность симуляции: оценивают базовые модели, наборы инструментальных средств, все возможные упрощения и допущения. Оценка рисков среды симуляции позволяет определить последствия неисправностей, неточностей или неполноты. Это позволит подтвердить или опровергнуть результаты, полученные в ходе симуляционного моделирования, например при сопоставлении таких результатов с реальными условиями. Например, модель, используемая для обоснования компонента функциональной безопасности, в задачах, связанных с распознаванием, позволит сделать выводы о реалистичности распознавания и сравнить с восприятием человеком и т. д. Для дополнительной информации см. 9.4.3;
- тип симуляции: ни один виртуальный инструмент тестирования не может быть использован для тестирования всех аспектов системы ИИ. По этой причине зачастую используется несколько инструментов для обеспечения уверенности в функциональной безопасности всей системы ИИ. Цепочка инструментов виртуального тестирования может включать следующие типы: модель в контуре (model in the loop, MIL), ПО в контуре (software in the loop,SIL), аппаратные средства в контуре (hardware in the loop, HIL);
- покрытие тестами: подход может включать случайную тестовую выборку, ограниченную тестовую выборку на основе определения входного пространства, тестовую выборку с распределениями на основе профиля пользователя, тестовую выборку критичности или важности на основе анализа функциональной безопасности, выборку стресс-теста на основе пограничных случаев или ожидаемых условий, которые могут создавать нагрузку на систему, и т. д. (см. *[4]*, С.5). Для многомерных входных данных это также может указывать на то, какая комбинация факторов проверяется;
- объем проведенных тестов: количество итераций симуляции, достаточное для обоснования аргумента функциональной безопасности.

Прежде чем инструменты виртуального тестирования можно будет использовать для верификации и валидации системы ИИ, необходимо провести верификацию и валидацию набора инструментальных средств. Выводы о наборе инструментальных средств для виртуального тестирования можно сделать, оценив четыре ключевые характеристики:

- соответствие цели: соответствие инструментов целям оценки системы ИИ. Соответствие цели предполагает четкое описание цели теста и определение всех граничных условий системы ИИ. Это включает в себя анализ операционной среды и определение требований для отдельных моделей. Сложность и уровень детализации каждой модели могут варьироваться в зависимости от актуальности, значимости и диапазона каждого фактора. Например, если условия эксплуатации исключают работу в ночное время, то модели датчиков не будут проверены в условиях низкой освещенности;
- возможности: способность виртуальных тестов выявлять сбои и связанные с ними риски скрытых ошибок. Возможности тестирования включают в себя определение допущений, ограничений и уровней точности инструментальных средств, способов оценки точности (КПЭ) и разумных допущений для КПЭ. Это может служить обоснованием того, что допущения, связанные с моделированием и реальной корреляцией, являются приемлемыми для цели тестирования. Обратите внимание, что выбранный уровень точности для моделей и возможные предположения играют важную роль для определения ограничений инструментальных средств;
- корректность (верификация): степень достоверности и надежности данных и алгоритмов инструментальных средств. Верификация позволяет изучить внедрение концептуальных или математических моделей, составляющих набор инструментальных средств. Это позволяет гарантировать, что инструментальные средства подходят для набора входных данных, которые нельзя протестировать на этапе валидации. Процедура основана на многоступенчатом подходе, который может включать верификацию кода, верификацию расчетов и анализ чувствительности;
- точность (валидация): способность виртуальных тестов воспроизводить целевые данные. Это включает в себя создание данных, которые можно использовать для демонстрации точности инструментов виртуального тестирования в сравнении с реальными условиями. Валидация инструментальных средств состоит из четырех основных этапов. Точная методология зависит от структуры и назначения инструментальных средств. Валидация может состоять из одного или нескольких этапов, описанных ниже:

a) валидация моделей подсистемы: например, моделей среды (инфраструктура, погодные условия, взаимодействие с пользователем), моделей датчиков [радар, камера, обнаружение света и определение дальности (лидар)], моделей шасси (привод, трансмиссия);

b) валидация системы шасси (модель шасси вместе с моделью среды);

c) валидация системы сенсора (модель сенсора с моделью среды);

d) валидация интегрированной системы (модель сенсора и модель среды под влиянием модели шасси).

Применение одних и тех же сценариев для инструментальных средств всех уровней (MIL, SIL и HIL) позволяет проводить валидацию системы и не требует проведения многочисленных физических тестов.

Использование инструментальных средств виртуального тестирования может зависеть от стратегий виртуальной валидации и верификации, реализованных во время их разработки. Таким образом, разработка модели и инструментальных средств обычно не стандартизируются, а скорее объясняются и анализируются в процессе сертификации.

Поэтому для общей оценки инструментальных средств виртуального тестирования требуется единый метод исследования таких свойств, позволяющий подтвердить данные, генерируемые такими инструментальными средствами. Важно исследовать влияние имитационных моделей и средств имитационного моделирования среди прочих инструментальных средств в случае возникновения сбоев, связанных с безопасностью, в конечном продукте. Предлагаемый подход к анализу критичности установлен в *ГОСТ IEC 61508-3.*

9.4.3 Рекомендации по проведению физического тестирования

Физическое тестирование может дополнять результаты тестирования методом моделирования. Тестирование системы в реальной среде или в конечной операционной среде позволяет обеспечить высокую достоверность результатов проверки реального использования. Для тестирования в реальных условиях рекомендации могут включать следующее:

- применение структурированных тестов, известных сценариев и тестов с вариантами использования. Например, тестирование автономных транспортных средств на тестовом полигоне или конкретные задачи на распознавание при тестировании сенсоров. Такие испытания хорошо представлены и позволяют отслеживать и сравнивать контрольные измерения во времени. Структурированные тесты могут включать разные типы входных данных, например анализ на уровне безопасности, технологии, продукта. Проведение комплексного тестирования требует глубокого понимания конечной цели применения;

- сочетание реальных испытаний с симуляционным моделированием. Физические тесты гораздо более ограничены в своей способности исследовать входное пространство из-за ограничений по стоимости и времени тестирования в сравнении с моделированием, но обеспечивают высочайшую точность реального использования и позволяют автоматически фиксировать случайные явления, которые невозможно смоделировать. В отличие от структурированных тестов, которые обычно проверяют «известное известное», как реальное, так и имитационное тестирование могут выявить различные типы «известного неизвестного»;

- связь реальных испытаний и симуляции. Реальные тесты позволяют проводить валидацию моделей, используемых в имитационных тестах;

- непрерывное тестирование и получение обратной связи. Тестирование в реальных условиях также позволяет установить «неизвестное». Отчеты об ошибках (и полученные данные о таких ошибках) предоставляют информацию для совершенствования сценариев моделирования;

- предметная область тестирования: границы эксплуатации. Определение предметной области тестирования (как при моделировании, так и при тестировании в реальных условиях) также важно, как и определение границ эксплуатации при тестировании в реальных условиях. Предметная область может включать ограничения на использование, ограничения среды, местоположения и времени, а также распределять обязанности между системой и пользователями и, при необходимости, другими системами. Кроме того, для оценки тестов важны показатели, показывающие охват такой области (это относится как к моделированию, так и к тестированию в реальных условиях);

- статистическая значимость. Процедуры и результаты тестирования основаны на надежных статистических принципах. Например, заключительная валидация функции аварийного останова проводится несколько раз, чтобы продемонстрировать, что соответствующие параметры находятся в заданных пределах, определенных с помощью статистического анализа. При этом верификация функции распознавания для обнаружения человека может выполняться на большой тестовой базе

данных, размер и охват которой определяются на основе заданной интенсивности отказов и доверительных интервалов.

9.4.4 Оценка уязвимости к аппаратным отказам

Уязвимость глубоких нейронных сетей к программным ошибкам низкая. Оценка доли отказов, приводящих к безопасному поведению (в отличие от небезопасного поведения), применима для определенных типов сетей. Возможные методы включают внесение неисправностей (например, отдельные веса в нейронной сети) в качестве агента ошибок в базовом аппаратном обеспечении. Например, можно проанализировать модели классификации, чтобы с уверенностью определить уязвимые части технологии ИИ к программным ошибкам.

9.5 Мониторинг и обратная связь

После одобрения и введения в эксплуатацию система ИИ будет одобрена и введена в эксплуатацию, статистика отказов системы ИИ позволит собирать данные об эффективности безопасности. Отчеты об отказах позволяют получать обратную связь с целью улучшения сценариев тестирования. Однако для основных функций, для которых невозможно получить абсолютное индуктивное или дедуктивное доказательство, приемлемые целевые показатели интенсивности отказов могут быть получены на основе целевых значений интенсивности отказов системы вместе с соответствующими обоснованиями для улучшения функциональной безопасности. При получении результатов эмпирическим способом методология испытаний и результаты также могут быть записаны.

Область проектирования системы на уровне операций и профилей использования в реальных условиях позволяет определить диапазон проблемы и создать показатели охвата тестирования (как при моделировании, так и в реальных условиях).

Статистическая значимость позволяет определить размер набора тестовых данных и тестовое покрытие на основе целевых частот отказов и доверительных интервалов, а также разработать рекомендации по приемлемым уровням достоверности.

Запись полевых тестов, если позволяют условия, может быть целесообразной для мониторинга показателей безопасности системы и выработки надлежащего ответа на отказы.

9.6 Объяснимый ИИ

Развивающаяся технология ИИ, известная как «объяснимый ИИ», связана с выявлением факторов, влияющих на принятие решений ИИ, чтобы процесс был понятен человеку, см. *[2]*. Достаточно объяснимая технология ИИ, если она будет успешно реализована, позволит разработчикам понимать алгоритмы принятия решений ИИ и предлагать способы для обеспечения функциональной безопасности алгоритмов МО в соответствии с действующими стандартами функциональной безопасности. В качестве альтернативы знания, получаемые человеком на основе наблюдений за моделью ИИ, могут быть повторно реализованы в виде традиционного ПО.

Хотя в настоящее время исследования объяснимости принятия решений для каждой системы ИИ класса II нецелесообразны, уже существуют некоторые подходы, позволяющие интерпретировать и объяснять структуры модели, которые могут помочь в процессах верификации и аудита. Например, тепловые карты внутренних узлов, влияющих на принятие конкретных решений, позволят понять причины таких решений. Эти техники, иногда называемые методом «серого ящика», позволяют понять поведение системы ИИ, особенно в тех случаях, когда процесс принятия решений системой не соответствует намерениям разработчиков. В таких случаях важно соотносить значения полученных объяснений с требованиями функциональной безопасности. Например, понимание процесса принятия решений, извлеченное из промежуточных слоев глубокой нейронной сети, само по себе не может быть достаточным для обоснования безопасного поведения, поскольку необъяснимые процессы на других уровнях могут свести ценность таких объяснений к нулю.

Дополнительную информацию об объяснимости ИИ см. в 8.3.

10 Меры контроля и снижения рисков

10.1 Введение

Создание устойчивой архитектуры системы ИИ, способной выдерживать сбои без потери свойств безопасности, важнее, чем улучшение качества ИИ. МО не влияет на разработку архитектуры безопасных систем, хотя и создает проблемы для обеспечения ее надежности и поведения при отказе.

Настоящий раздел посвящен методам усовершенствования моделей МО как компонентов систем ИИ и вариантам использования окружающих их подсистем для улучшения нефункциональных свойств, таких как надежность, доступность и качество. В 10.2 описаны ключевые вопросы, связанные с архитектурой подсистемы ИИ, включая методы смягчения последствий и контроля. В 10.3 представлены методы повышения надежности компонентов. Механизмы отказа, описанные в разделе 8, указывают на различные вызовы для отдельных компонентов моделей МО. Раздел 9 посвящен процессам верификации и валидации этих компонентов на протяжении всего жизненного цикла. Меры, описанные в настоящем разделе, направлены на понимание режимов отказа и являются частью обеспечения устойчивости процесса МО, описанного в разделе 11.

10.2 Архитектура подсистем ИИ

10.2.1 Введение

Оценка безопасности на системном уровне позволяет определить уровень надежности подсистемы функции, включающей компоненты МО. Архитектура подсистемы может включать дополнительные технологии для удовлетворения этих требований. Анализ следующих особенностей позволит выработать возможные решения:

a) безопасная (субоптимальная) функция, которая может быстро заменить в процессе работы системы компоненты на основе МО, может быть разработана с использованием методов, не использующих ИИ. Такой резервной системой принятия решений может быть, например, альтернативный контроллер или отказоустойчивое нулевое действие (например, отключение питания). Резервное действие позволяет использовать методы обнаружения для переключения при обнаружении небезопасных условий;

b) безопасное подмножество пространства действия может быть определено (заранее или онлайн) с помощью супервизора с ограничениями;

c) избыточное число моделей МО, снабженное схемой голосования и агрегации для получения финального решения.

Примечание — Предлагаемые архитектурные решения могут применяться совместно или быть альтернативными, т. е. одного из них может быть достаточно.

Включение технологии ИИ создает определенные проблемы для каждого из вариантов архитектуры. В 10.2.2 описаны механизмы обнаружения аномального ввода, вывода или внутреннего состояния (например, сила активации нейрона), которые могут применяться для выявления возможных отказов. В 10.2.3 описана функция контроля с использованием элементов теории управления для сведения работы ИИ к минимуму. В 10.2.4 показаны различные способы обеспечения избыточности с помощью технологий ИИ. 10.2.5 посвящен вопросу проектирования системы ИИ с функцией статистического оценивания.

10.2.2 Механизмы отслеживания момента для переключения

Архитектура, представленная на рисунке 5, часто характеризуется как состояние нагруженного резерва в отказоустойчивых системах. Например, компонент супервизора — диспетчер, производящий мониторинг, способен отследить потенциально небезопасные действия, совершаемые ИИ из-за внутренних и внешних сбоев. Обнаружение таких действий позволяет предпринять шаги для поддержания системы в безопасном состоянии. Мониторинг может быть разработан как с использованием технологии ИИ, так и без нее. В первом случае необходимо убедиться, что мониторинг и первичная система независимы друг от друга.

Рисунок 5 — Архитектурные паттерны для систем, использующих компоненты ИИ

Приемлемость поведения технологии ИИ можно оценить на основе ее обучающих данных [рисунок 6 a)]. Обычно невозможно проверить поведение модели в случае ранее не встречаемых входных данных вне распределения (out-of-distribution, OOD), что является следствием дрейфа данных. Приемлемые выходные данные модели основаны на непроверенных свойствах обобщения модели и могут быть ошибочными, поэтому важно обнаружить такие отличающиеся входные данные. Распределение может зависеть как от параметров в отдельном образце, так и от их изменения во времени (т. е. динамики). Простое ограничение допустимых входных данных не позволит обнаружить пробелы в обучающих данных, особенно в случае многомерных систем. При отсутствии улучшений обучающих данных методы обнаружения в некоторых случаях могут быть выбраны на основе свойств данных (размерности, линейности корреляций параметров, динамики, сезонный дрейф и т. д.). Однако состязательные методы продемонстрировали чрезвычайную чувствительность глубоких сетей к небольшим, казалось бы, случайным отклонениям (например, неверная классификация изображений путем добавления шума, что затрудняет надежный анализ входных данных). Состязательные методы включают проверку времени выполнения и проверку входящих данных для обнаружения состязательных атак.

В отличие от мониторинга входных данных [рисунок 6 a)] мониторинг выходных данных позволяет обнаружить нежелательное поведение, возникающее в результате OOD или находящихся в распределении входных данных. Мониторинг выхода за известную границу [рисунок 6 b)] или альтернативная модель описаны в литературе, посвященной вопросам обнаружения неисправностей (например, статистическое обнаружение или обнаружение остатков на основе моделей). Границы также могут адаптироваться к последовательности многовариантных входных сигналов системы и, таким образом, не позволять системе покидать область безопасного состояния [рисунок 6 c)]. Для динамических систем решение об обнаружении также гарантирует, что безопасное состояние достижимо без нарушения ограничений (т. е. инерция или нестабильность системы не препятствуют входу в опасные состояния по решениям резервного контроллера). Там, где выходные распределения не поддаются традиционным методам моделирования, может применяться МО для создания монитора. Одним из примеров является обучение (более простой) вторичной сети (например, архитектуры «ученик — учитель») на выходных данных, сгенерированных моделью МО, и использование их с помеченными данными для прогнозирования достоверности выходных данных.

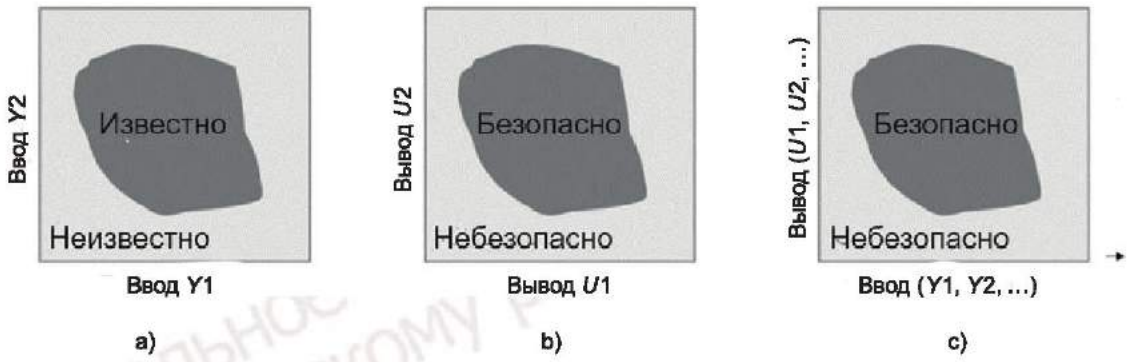


Рисунок 6 — Оценка приемлемого поведения технологии ИИ

Также можно использовать метаданные модели, такие как активация внутренних нейронов или посредством расчета меры неопределенности. Меры неопределенности в МО могут быть получены точным выводом (байесовская нейронная сеть), рандомизация может быть имплементирована и приблизительно (прореживание, ансамбли, использование сотфмакса в выходном слое сети). Уверенность в выходных данных (уровень активации) может быть полезна, но требует тщательной калибровки вероятности и подвержена рискам. Эти риски включают чрезмерную уверенность (классификаторы часто дают сбой, предоставляя неправильные, но уверенные ответы) не только в крайних случаях или за пределами обучающего пространства, но также и при небольших неисправностях, как в случае состязательных примеров.

При разработке мониторинга необходимо принимать во внимание четыре аспекта:

- тип ошибок ИИ, которые могут быть обнаружены;
- способы выявления сбоев ИИ во время выполнения задач;
- бенчмарк-тестирование разных способов мониторинга;
- типы вмешательств, которые позволяют обойти неисправность после ее обнаружения или обойти потенциальную опасность.

Надстройки для оценки неопределенности позволяют оценить качество решения и также могут применяться либо для автоматического принятия решений, либо в качестве входных данных для оценки правильности предложенного системой ИИ решения человеком.

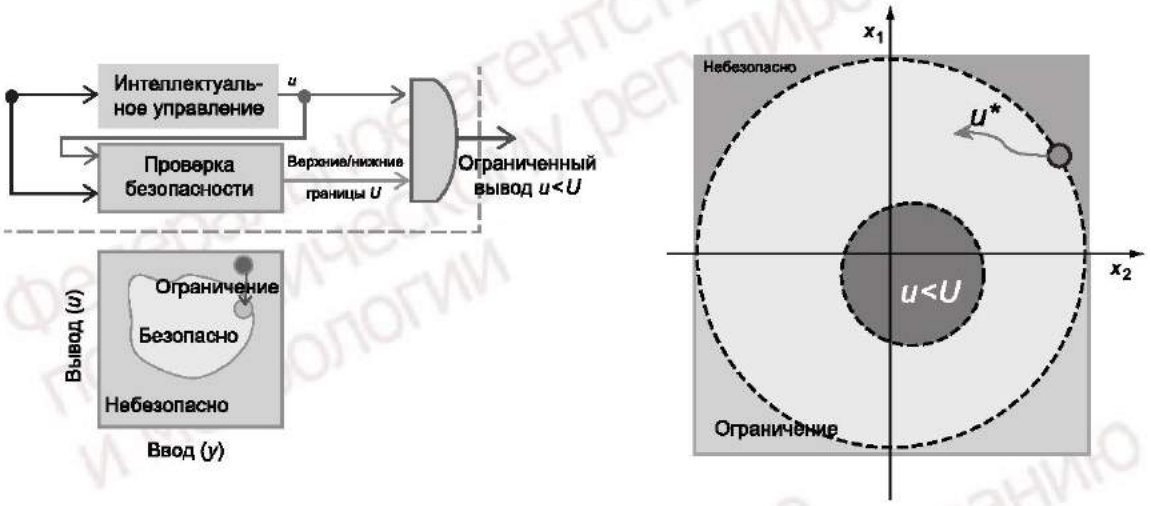
10.2.3 Использование функции наблюдения с ограничениями для управления поведением системы в безопасных пределах

При наличии соответствующего модуля контроля работы система ИИ может быть ограничена заранее определенными пределами безопасности. Пределы безопасности позволяют определить подмножество пространства действий (безопасная зона) и минимально ограничить безопасное поведение компонентов МО. Однако простые ограничения вывода могут чрезмерно подавлять компонент МО, что приводит к поведению, имитирующему сам ограничитель, и сводит на нет преимущества реализации компонента МО. Эту архитектуру подсистемы иногда называют каркасом безопасности, который навязывает поведение подсистеме.

Например, на рисунке 7 a) система ИИ может использоваться как часть интеллектуального управления для обеспечения оптимального решения. В этой архитектуре функция безопасности без ИИ выводит допустимый диапазон выходных данных для заданного входного сигнала и ограничивает выходной сигнал интеллектуального управления.

Ограничение вывода на основе функции ввода обычно не подходит для систем с динамикой, где состояние системы (x) определяет небезопасные области, но не реагирует мгновенно на изменение входных данных контроллера. Формально минимальные границы могут быть определены для динамических и гибридных систем с помощью методов теории управления, таких как подходы барьерных функций, которые позволяют определить инвариантный набор под управлением системы без ИИ, см. рисунок 7 b). Эти подходы позволяют гарантировать, что система не выйдет за операционную область, определенную как безопасная, для некоторого ограниченного входного управляющего сигнала (подмножество u всех возможных управляющих сигналов U). Эти наборы, как правило, консервативны, поскольку они не нацелены на активное восстановление системы и ее возврат в более безопасные области, поэтому функции управляющего барьера могут применяться в качестве механизмов обнару-

жения для переключения в обычные стабилизирующие средства управления (генерирующие управляющий сигнал u^*, если они доступны в условиях подходящего контроля ИИ, как описано в 10.2.1). Для систем со сложными требованиями безопасности, например многоканальных измерительных систем, в которых используется технология ИИ, можно рассмотреть такие функции проверки, как метрологический самоконтроль или самовалидация.



a) через контролирующие ограничения выходного сигнала

b) на непрерывном выходе управления через барьерные сертификаты

Рисунок 7 — Безопасное применение технологии ИИ для управления

10.2.4 Избыточность, концепция ансамбля и разнообразие

Избыточность бывает разных видов: структурная (пространственная), временная (частотная), функциональная (информационная, аналитическая) или комбинированная. Например, при использовании нейронных сетей избыточность для технологий ИИ включает:

- аналитическую избыточность: методы обнаружения и изоляции (failure detection and isolation, FDI) основаны на сравнении доступных измерений системы с априорной информацией, представленной математической моделью системы. В этом подходе наблюдаются две основные тенденции, а именно: аналитическая избыточность или остаточная генерация и оценка параметров;
- множественные вычисления с избыточностью по времени: например, параллельное исправление ошибок возможно при использовании избыточности по времени на основе повторных вычислений и тройного резервирования с голосованием (recomputing with triplication with voting, RETWV);
- n-версионное программирование: в этом методе несколько симплексных моделей обучаются независимо, поэтому маловероятно, что эти модели дадут ошибочные результаты для одних и тех же тестовых случаев. Таким образом, разработка отказоустойчивой системы, выход которой определяется всеми этими моделями вместе, возможна;
- использование избыточных глубоких архитектур;
- ансамблевое использование нейронных сетей для создания надежных классификаторов: идея состоит в том, чтобы объединить несколько «слабых» классификаторов с целью получить «сильный» классификатор, который сможет обеспечить надежную работу при выходе из строя одного из его членов;
- использование основанной на алгоритмах отказоустойчивости для нейронных сетей.

Методы метрологического самоконтроля (включая самовалидацию и самодиагностику), применяемые для систем контроля приоритетного оборудования, могут быть адаптированы для технологии ИИ.

Для повышения эффективности избыточность сочетается с разнообразием, что позволяет снизить вероятность систематических сбоев во время разработки. Это связано с несколькими технологиями ИИ, демонстрирующими одинаковое поведение, но реализованными:

- разными командами;
- с использованием разных правил разметки;
- с использованием разных формулировок задач;

- с использованием разных обучающих данных;
- при работе на разном оборудовании (также для отказов, не связанных с ИИ);
- при разнообразии сенсоров, генерирующих данные;
- при разных методах самоконтроля и самовалидации;
- в условиях разнообразия самой технологии ИИ.

Ввиду сложной и неопределенной природы разнообразие может быть выражено множеством показателей. Необходимо тщательно оценить все эти показатели, чтобы найти ответы на следующие вопросы: в какой степени технологии ИИ при одинаковых условиях обучения могут различаться по своей производительности и надежности? Подходят ли показатели разнообразия для выбора участников при формировании более надежного ансамбля?

Другой возможный подход заключается в выявлении и устранении ложных обнаружений путем сравнения решений по ключевым точкам из разных нейронных сетей. Эта форма разнообразного сравнения часто сочетается с мониторингом.

Если рассматривать избыточность как часть обоснования безопасности и принимать во внимание объяснимость глубинных нейронных сетей, нужно контролировать отсутствие общей возможной причины отказа сразу всех отдельных нейронных сетей, входящих в ансамбль.

10.2.5 Система ИИ со статистической оценкой

После обучения многие системы ИИ становятся детерминированными в выводах, однако там, где входная размерность высока и непрерывна по значению, производительность этих систем обычно характеризуется статистически. Например, кросс-валидация нейронной сети может давать разные показатели производительности в зависимости от изменяющихся выборок распределения обучающих данных в каждой «кратности» проверки. Для конкретного приложения в определенных условиях эксплуатации система, содержащая технологию ИИ, может быть оценена на предмет функциональной безопасности с учетом статистического распределения ее выходных данных. При невозможности получения жесткой верхней границы частоты ошибок возможно определить статистический доверительный интервал максимальной ошибки для заданного входного распределения. Ключевое допущение состоит в том, что эти статистические данные основаны на распределении тестовых данных, схожих с производственными данными.

При условии, что некоторые события имеют низкую вероятность, но большое влияние, целесообразно увеличить частоту этих событий во входных данных, сделав частоту событий во входных данных пропорциональной риску, а не фактической вероятности возникновения.

Подход к оценке может быть основан на следующих шагах:

- анализ технологии ИИ, например модели МО;
- анализ системы ИИ как обычной математической модели с вероятностным поведением.

Чем более гибкой является модель, тем сложнее ее анализировать.

Для технологий ИИ класса II и класса III крайне важно принимать во внимание статистическую информацию.

10.3 Повышение надежности компонентов с технологией ИИ

10.3.1 Общая информация о методах компонентов ИИ

Чтобы повысить устойчивость к шумовым помехам, сбоям устройств и, возможно, злонамеренным (враждебным) действиям, можно использовать несколько методов как на этапах тестирования, так и на этапах обучения. Такие методы включают:

- регуляризацию — это методология, позволяющая смягчить проблему переобучения и, таким образом, повысить стабильность. Этот метод можно считать аналогичным методам, используемым в регрессионной подгонке, где величина веса или ненулевые значения функции потерь при обучении исключаются или задаются априорным распределением. Такой подход предпочтительнее, чем удаление факторов с весами с низким значением после обучения. Альтернативные методы включают конфигурацию архитектуры нейронной сети с общими весами на некоторых соединенных узлах (например, на повторяющихся элементах фильтра в СНС) для упрощения структуры модели. Прореживание считается хорошей практикой для уменьшения переобучения в глубинных нейронных сетях за счет дополнительного гиперпараметра «уровень дропаута» («dropout rate»). Этот метод случайным образом отключает элементы сети на небольшой части обучения. Поскольку сеть не может полагаться исключительно на один узел для моделирования конкретной функции данных, исключенные регионы не имеют избыточной специализации;

- обучение с учетом сбоев, включающее моделирование ошибок во время обучения нейронной сети, которое может сделать нейронные сети более устойчивыми к конкретным моделям сбоев на устройстве в случае, когда нарушения системы ИИ предсказуемы (например, для аппаратных ошибок);

- состязательное надежное обучение — это метод обучения, который минимизирует или ограничивает наихудшие погрешности при данных обучения, дополненных моделью возможных отклонений при атаках. Одновременная максимизация состязательного эффекта и минимизация ошибок приводит к расширению стандартных алгоритмов обучения градиентного спуска. Другие подходы могут обеспечить надежность неизменности вывода, например, формально доказывая, что изменения классификации невозможны, если отклонения находятся в заданных пределах. Однако масштабирование этих подходов на крупномасштабные и гетерогенные сети остается проблемой;

- подходы рандомизации, такие как рандомизированное сглаживание, представляющие собой эффективный эквивалент обучению с несколькими режимами данных, дополненных рандомизированным шумом, с целью вычисления конечного значения результата как среднего значения по отношению к распределению шума;

- робастность к входным данным вне распределения для приложений, подверженных работе с ограниченными обучающими данными, смещению данных или концепций. Расширение и обогащение данных сокращает расстояния, которые система ИИ должна экстраполировать из обучающих данных. Например, более высокой производительности можно достичь при перемещении и разворачивании изображения в обучающих данных. Такое обучение дает дополнительный эффект и повышает робастность.

10.3.3 Технологии оптимизации и сжатия

Технологии оптимизации и сжатия, такие как квантование параметров и вычислений (т. е. сокращение диапазона параметров), сокращение избыточных нейронов в скрытых слоях (т. е. удаление менее важных параметров из модели) и дистилляция в более простые суррогатные модели, позволяют обеспечить вторичные преимущества для системы. Как и при всех модификациях системы, необходим тщательный анализ рисков потери производительности.

Сокращение размерности входных данных с помощью методов, не связанных с ИИ (линейные и нелинейные основные компоненты, кластеризация, извлечение признаков и т. д.), создает риски безвозвратного исключения полезной информации. Потенциальной альтернативой, часто используемой в современных нейросетевых архитектурах, является использование встраиваемых слоев (таких, как, например, сверточный слой или низкоразмерные полносвязные узлы). Это приводит к дальнейшему упрощению, а также часто повышает эффективность обучения.

Упрощенные модели с уменьшенной размерностью весов (и, возможно, входных данных) могут упростить обучение и снизить риск невыпуклости (и, следовательно, множественных локальных минимумов) в ландшафте функции потерь. Помимо возможности улучшения конвергенции, уменьшенные размеры сети интуитивно упрощают интерпретацию. Однако даже уменьшенный размер сети все же может превысить способность понимания функции каждого параметра по отношению к его вкладу в итоговый ответ модели (т. е. его прослеживаемость). Новые методы визуализации дали многообещающие результаты, особенно в области классификации изображений, но их полнота еще не доказана. Модульность сети — это еще один прагматичный способ помочь понять ее прослеживаемость и упростить проверку (включая возможность сделать формальные подходы к проверке вычислительно осуществимыми).

Дистилляция знаний изначально была предназначена для создания более простых суррогатов и более удобных для вычислений моделей. Эта концепция создает вторичную, более простую модель, обученную на выходе более крупной модели, а не на обучающих данных. Сложная модель обычно создает вложение меньшего размера, чем необработанные данные, и после изучения часто может быть смоделирована с помощью вторичных линейных моделей с незначительной потерей производительности и преимуществом объяснимости. Нелинейные вторичные модели в меньшей степени способствуют интерпретируемости, но могут помочь в сглаживании градиентов. Сглаженные градиенты могут обеспечить маскирование градиента для защиты от атак злоумышленников, уменьшая изменение выходных данных для данного входного отклонения. Это достигается созданием вероятностных меток на первом пути обучения со сложной моделью, а затем сохранением более простой модели с этими «нечеткими» вероятностными метками.

Обрезка некоторых нейронов может защитить от атак во время обучения. Один из подходов к сокращению сети достигается путем анализа активации нейронов с чистыми входными данными после

обучения, итеративным удалением данных с низкой активацией и повторным тестированием. Это снижает риск обнаружения уязвимости сети.

10.3.4 Механизмы внимания

Целью механизма внимания является повышение производительности прогнозирования в моделях последовательностей, таких как модели языкового перевода, преобразователи речи в текст и модели для создания субтитров к изображениям. Существует несколько рекомендаций, связанных с механизмами внимания:

- механизм внимания для изучения глобального контекста: механизм внимания изучает взаимосвязь между последовательностью признаков (например, словами в предложении), используя взвешенную комбинацию всех закодированных входных векторов. Точно так же в приложениях машинного зрения веса внимания могут изучать глобальное взвешивание по всему изображению для решения более сложных задач, таких как создание подписей к изображениям, на которые сверточные слои не способны. Позже были предложены трансформеры для изображений для захвата контекста в изображениях без каких-либо последовательных данных (например, текста), доступных для обучения. Наконец, механизм внимания используется как подходящее решение для обучения моделей на мультимодальных данных, особенно на комбинациях последовательных и пространственных данных;

- карты внимания для проверки работоспособности и манипулирования функциями: карта внимания (также известная как карта заметности или карта чувствительности) — это распространенный тип объяснения МО с целью указания на наиболее важную функцию в данном прогнозе. Карта внимания — это тип локального объяснения, которое ограничено предсказаниями отдельных моделей независимо от общего поведения модели, но все же подходит для исследования пограничных случаев при отладке модели. Карты внимания могут быть получены различными способами, такими как локальная аппроксимация глубоких моделей с использованием неглубоких интерпретируемых моделей. Обратите внимание, что карты внимания могут быть либо априорными, либо интегрированными в сеть;

- преимущества карт внимания: просмотр объяснений МО дает разработчикам преимущества для улучшения данной модели на нескольких этапах жизненного цикла МО. Например, на этапах выявления проблем в структуре модели, проектирования на основе признаков и улучшения обучающих данных. Кроме того, если предположить, что объяснения модели согласуются с рассуждениями и пониманием данных конечным пользователем, просмотр карт внимания человеком способствует созданию соответствующего уровня доверия к автономным системам;

- тренируемое внимание: механизмы тренируемого внимания имеют веса внимания, которые изучаются во время обучения для повышения эффективности внимания. Модуль множественной оценки внимания может быть использован для разных сетевых уровней в целях создания более точных карт внимания и более высокой производительности прогнозирования. Выраженный надзор человека (например, отслеживание взгляда) для моделей внимания также может быть использован, но этот подход требует больших затрат на разметку данных;

- достоверность объяснения: поскольку объяснения моделей всегда являются неполными объяснениями моделей «черного ящика», на достоверность и полноту объяснений сильно влияют такие факторы, как эвристический метод, входной пример, размер и качество обучающих данных.

10.3.5 Защита данных и параметров

Данные и параметры модели потенциально уязвимы для случайных и преднамеренных нарушений и потерь, что может быть вызвано отказом оборудования или отравлением данных в результате атак злоумышленников. Как и в случае со всеми данными, используемыми в системе, использование процессов оценки рисков и управления данными (см. раздел 11) позволяет управлять используемыми мерами защиты с учетом конкретных проблем, связанных с технологией ИИ с интенсивным использованием данных (например, объем, разнообразие, скорость, изменчивость).

Управление конфигурацией данных поддерживается на протяжении всего жизненного цикла модели, включая происхождение, права доступа и показатели качества данных. Можно использовать процесс настройки, приведенный в *ГОСТ Р ИСО 26262-6—2021* (приложение С). Обеспечение информации о данных во время выполнения и автономного обучения должно учитывать множество свойств: целостность, полноту, точность, разрешение и т. д.

В дополнение к мерам контроля данных разумной предосторожностью является предварительная обработка потока входных данных для удаления недопустимых входных шаблонов. Например, фильтры, прозрачные для физической полосы пропускания системы и способные удалять нежелательный шум, могут дополнять механизмы обнаружения, предотвращая передачу данных вне распределения.

Высокие вычислительные требования МО побуждают разработчиков использовать сторонние высокопроизводительные вычисления для обучения модели, где обеспечение достоверности информации сопряжено с более высоким риском. Уменьшение утечки интеллектуальной собственности не будет достаточным (например, шифрование, обфускация меток и распространение данных). Манипулирование обучающими данными (отравление) может быть разработано злоумышленником для обхода локального тестирования, когда сеть ведет себя правильно на обычных тестовых данных с неактивными проблемами (например, нейроны, не активированные при обычном обучении). Дополнение методов сокращения переобучением (облегченным) на локально защищенном источнике данных может помочь снизить чувствительность к состязательным примерам.

11 Процессы и методология

11.1 Общие положения

С точки зрения функциональной безопасности многие проблемы жизненного цикла являются общими для систем с ИИ и без него. Их общие черты описаны в 11.2.

Уровень функциональной безопасности, которого должна достичь система, не зависит от того, используется технология ИИ или нет. Части системы, построенные с использованием программных подходов, не связанных с ИИ, могут обрабатываться в соответствии с существующими стандартами функциональной безопасности. Методология, обычно используемая для разработки моделей ИИ, имеет неотъемлемые пробелы с точки зрения требований стандартов функциональной безопасности. Поиск подходящих средств для устранения «пробелов» является важной задачей.

Для обеспечения функциональной безопасности важно учитывать безопасность на протяжении всего жизненного цикла системы.

11.2 Связь между жизненным циклом ИИ и жизненным циклом функциональной безопасности

Традиционно термин «жизненный цикл» используется в нескольких целях. Одной из целей является предоставление определенного набора процессов в рамках жизненного цикла системы, аппаратного или ПО, а также облегчение взаимодействия между заинтересованными сторонами этого жизненного цикла. *[2]* описывает высокоуровневую модель жизненного цикла систем ИИ, а [1] определяет процессы жизненного цикла систем ИИ. Процессы жизненного цикла ПО также описаны в *[16]*.

Дополнительная цель — уточнить, что должно быть достигнуто на каждом этапе на протяжении всего жизненного цикла для реализации определенного уровня функциональной безопасности. Это установлено в *ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ Р МЭК 61508-4* и других стандартах функциональной безопасности. Цель в данном случае состоит в том, чтобы прояснить, что должно быть достигнуто. С этой целью в серии стандартов *ГОСТ Р МЭК 61508* определен жизненный цикл функциональной безопасности, который включает этап анализа опасностей и рисков и этап распределения общих требований функциональной безопасности, что было описано в общих требованиях, см. *ГОСТ Р МЭК 61508-1*. Дополнительные специфические требования для аппаратного обеспечения установлены в *ГОСТ Р МЭК 61508-2*, а для ПО — в *ГОСТ IEC 61508-3*.

В настоящем стандарте указано, что разумно начинать с традиционного жизненного цикла функциональной безопасности, а затем модифицировать и адаптировать жизненный цикл функциональной безопасности с учетом проблем, характерных для системы ИИ, которые имеют влияние на функциональную безопасность. Этап анализа опасностей и рисков может основываться на серии стандартов *ГОСТ Р МЭК 61508* или других стандартах функциональной безопасности, адаптированных для учета специфических особенностей ИИ, перечисленных в [1] в качестве свойств, важных для функциональной безопасности (см. раздел 8).

Серия стандартов *ГОСТ Р МЭК 61508* и другие стандарты функциональной безопасности ссылаются на модель V как основу жизненного цикла, хотя в некоторых стандартах (например, *ГОСТ IEC 61508-3, ГОСТ Р МЭК 61508-1, ГОСТ Р МЭК 61508-2, ГОСТ Р МЭК 61508-4, ГОСТ Р МЭК 61508-7* и *ГОСТ Р МЭК 61511-1*) указано, что жизненный цикл или его этапы могут быть адаптированы к конкретной технологии реализации.

Жизненный цикл функциональной безопасности для разработки системы ИИ должен быть определен во время планирования функциональной безопасности (см. рисунок 8).

Допустимо адаптировать V-модель для моделей поэтапной разработки, чтобы она соответствовала особенностям ИИ, например, как показано в [1]. При выполнении итеративной и поэтапной разработки (например, циклов итеративного обучения) важна регрессионная валидация.

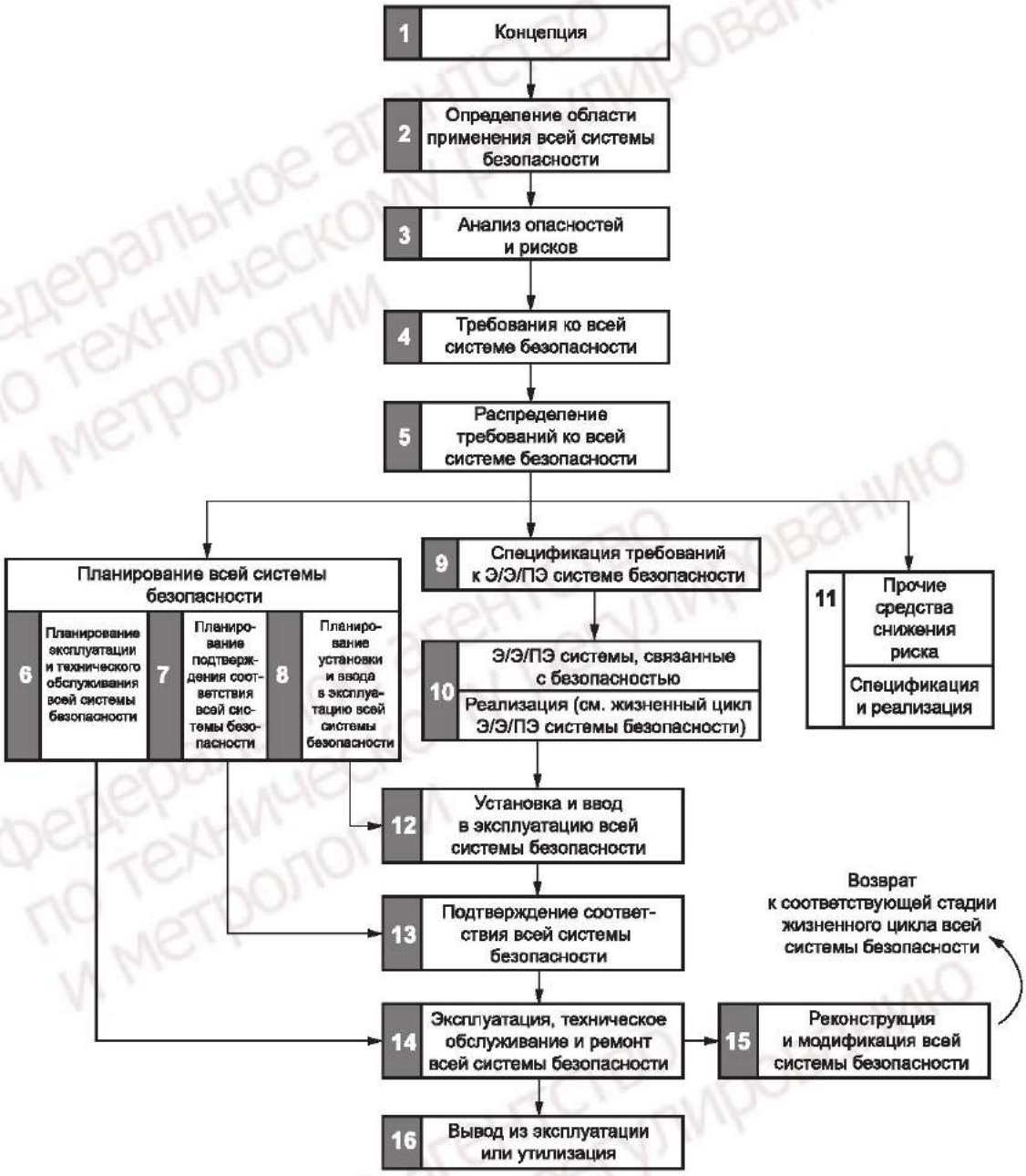


Рисунок 8 — Модель жизненного цикла (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2)

11.3 Этапы ИИ

Пример соответствия *[1]* и *ГОСТ Р МЭК 61508-1* приведен в приложении D.

11.4 Документация и артефакты функциональной безопасности

Достаточность информации и документации для каждой фазы жизненного цикла функциональной безопасности способствует последующим фазам и действиям по проверке. Это включает в себя документирование изменений в продуктах и процессах.

Проблемы, характерные для систем ИИ, включают процессы обучения, актуальность данных и достаточность документации данных обучения, валидации и тестирования.

11.5 Методология

11.5.1 Введение

В настоящем подразделе описана наиболее важная методология для технологий ИИ.

11.5.2 Модели отказов

Концепция моделей отказов предназначена для систематического и, возможно, автоматизированного анализа поведения элемента при отказах. Сущность модели отказов состоит в том, чтобы охватить многочисленные детали реальности достаточно высоким уровнем абстракции. Особенно в области МО крайне важно повышать осведомленность об отказах: модели являются ключом к достижению этой цели.

Модель отказов представляет собой упрощенную абстракцию реальных эффектов, которые могут вызвать ошибки, предназначенную для систематического анализа. Часто разные эффекты связаны с одним отказом. На самом деле распространение отказа довольно сложное, но часто разные цепочки распространения приводят к одинаковым отказам.

При достаточно точном определении влияние отказов можно смоделировать или проанализировать вручную. Это необходимо для оценки функциональной безопасности. Применение модели отказов ко всем элементам позволяет добиться полноты относительно уровня абстракции модели отказов.

Для создания модели отказов необходимо описать и спроектировать систему с соответствующими элементами. Для каждого из этих элементов определяются виды отказов с помощью метода управляющих слов, такого как анализ опасности и работоспособности (hazard and operability analysis, HAZOP).

Для МО к моделям отказов относят следующие аспекты:

- наборы данных, используемые для обучения, валидации и тестирования;
- модель МО;
- процесс обучения;
- связь жизненного цикла МО с жизненным циклом безопасности [в том числе возможность FMEA процесса (PFMEA)].

Аспекты валидации и верификации описаны в разделе 9.

11.5.3 PFMEA для автономного обучения технологии ИИ

FMEA может применяться на уровне процесса, на функциональном уровне или на уровне элементов, например на этапе автономного обучения системы ИИ.

PFMEA можно использовать для анализа и устранения возможных источников систематической ошибки и ограничений в процессе автономного обучения (например, во время обучения модели МО перед этапом развертывания). Также можно рассмотреть дополнительные методы анализа, такие как классификационный FMEA (CFMEA), который представляет собой метод, специализированный для оценки восприятия на основе классификации.

Приложение А
(справочное)

Применимость *ГОСТ IEC 61508-3* к элементам технологии систем ИИ

А.1 Введение

Целью данного приложения является показать на примере, в какой степени и какими средствами можно использовать методы, приведенные в *ГОСТ IEC 61508-3*. Приложение А [а также соответствующие таблицы *ГОСТ IEC 61508-3—2018* (приложение В) с описаниями по *ГОСТ Р МЭК 61508-7—2012* (приложения В и С)] может применяться с элементами технологии систем ИИ, которые могут быть признаны соответствующими стандартам функциональной безопасности.

Примечание — В отношении схемы классификации, описанной в разделе 6, настоящее приложение применяется к элементам технологии ИИ класса I, а приложение В — к элементам класса II.

А.2 Анализ применимости методов/средств, установленных *ГОСТ IEC 61508-3—2018* (приложения А и В), к элементам технологии систем ИИ

В таблицах А.1—А.19 представлен подход к интерпретации таблиц по *ГОСТ IEC 61508-3—2018* (приложения А и В) для элементов технологии системы ИИ, которые могут быть признаны соответствующими стандартам функциональной безопасности.

Примечание — В таблицах А.1—А.19, ссылки в колонке «Метод/средство» относятся к пунктам *ГОСТ IEC 61508-3—2018*, а ссылки «В.х.х.х» и «С.х.х.х» во второй колонке каждой таблицы (с заголовком «Ссылка») указывают на подробное описание методов и средств, приведенных в приложениях В и С *ГОСТ Р МЭК 61508-7—2012*.

Таблица А.1 — Интерпретация спецификации требований к ПО системы безопасности (*ГОСТ IEC 61508-3—2018*, таблица А.1)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1a	Полуформальные методы	Таблица А.17	Полуформальные методы используются для описания архитектуры МО в виде блок-схем, описания слоев, связей и поведения входных потоков
1b	Формальные методы	В.2.2, С.2.4	
2	Прямая прослеживаемость между требованиями к системе безопасности и требованиями к ПО системы безопасности	С.2.11	Для элементов технологии, не зависящих от варианта использования: применимо, как и для элементов системы без ИИ. Для элементов технологии, зависящих от случая использования, в некоторых случаях может быть трудно определить спецификацию требований безопасности для модели ИИ. Например, требованием безопасности может быть обнаружение всех пешеходов на дороге, но как четко определить все возможные случаи для пешеходов (например, человека на инвалидной коляске)? С другой стороны, *[17]* и *[18]* определяют функцию обнаружения человека, которая может быть разложена на программные функции и отслежена. Например, определенное количество пикселей или образцов измерения возвращает значение с заданным допуском. Такой подход может быть использован независимо от базовой программной технологии, включая технологию ИИ
3	Обратная прослеживаемость между требованиями к системе безопасности и предполагаемыми потребностями безопасности	С.2.11	Также применимо к элементам технологии ИИ
4	Компьютерные средства разработки спецификаций для поддержки подходящих методов/средств в таблице А.1	В.2.4	Также применимо к элементам технологии ИИ

Таблица А.2 — Интерпретация проектирования и разработки ПО: проектирование архитектуры ПО (*ГОСТ IEC 61508-3—2018*, таблица А.2)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Обнаружение ошибок	С.3.1	Существует несколько возможных методов обнаружения неисправностей ИИ как во время выполнения (вывод), так и в автономном режиме (обучение), включая: - проверку оперативной области на наличие сдвигов в распределении; - проверку новых концепций (например, новые объекты, другое поведение, новые правила); - изменения, происходящие в мире (смещение домена, новые объекты, изменение правил). Таким образом, различают обнаружение ошибок во время обучения и во время вывода
2	Коды обнаружения ошибок	С.3.2	Также применимо к элементам технологии ИИ
3a	Программирование с проверкой ошибок	С.3.3	Также возможно для элементов технологии ИИ
3b	Методы контроля (при реализации процесса контроля и контролируемой функции на одном компьютере обеспечивается их независимость)	С.3.4	Также возможно для элементов технологии ИИ: монитором может быть как традиционно разработанный механизм, так и другая технология ИИ (например, обученная по-другому или реализующая другой алгоритмический подход ИИ) или наличие n-модульной архитектуры с различными глубокими нейронными сетями, решающими одну и ту же задачу и голосующими. Учитывается не только разнообразие между ПО и алгоритмом ИИ, но и разнообразие между данными, на которых обучается алгоритм МО. Для ПО актуально также учитывать разнообразие аппаратного обеспечения, которое может включать разнообразие в реализации ПО нижнего уровня, разнообразие скомпилированных инструкций, выполнение инструкций и т. д. Использование разнообразных методов более подробно рассматривается в 10.2.4
3c	Методы контроля (реализация процесса контроля и контролируемой функции на разных компьютерах)	С.3.4	
3d	Многовариантное программирование, реализующее одну спецификацию требований к ПО системы безопасности	С.3.5	
3e	Функционально многовариантное программирование, реализующее различные спецификации требований к ПО системы безопасности	С.3.5	
3f	Восстановление предыдущего состояния	С.3.6	Также используется для технологии ИИ в целом (при условии достаточного пространства для хранения состояния) и может повысить робастность результата ИИ, поскольку такая методология вводит своего рода избыточность (небольшие изменения во входном векторе)
3g	Проектирование ПО, не сохраняющего состояние (или проектирование ПО, сохраняющего ограниченное описание состояния)	С.2.12	Не подходит для элементов технологии ИИ
4a	Механизмы повторных попыток парирования сбоя	С.3.7	Также используется для технологии ИИ в целом (при условии достаточного пространства для хранения состояния) и может повысить робастность результата ИИ, поскольку такая методология вводит своего рода избыточность (небольшие изменения во входном векторе)
4b	Постепенное отключение функций	С.3.8	Для элементов технологии ИИ может применяться постепенное отключение функций в случае снижения уверенности в выходном значении
5	Исправление ошибок методами ИИ	С.3.9	—

Окончание таблицы А.2

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
6	Динамическая реконфигурация	С.3.10	Это важно для будущих открытых систем, помимо ИИ. Существуют различные мнения, основанные на конкретном элементе системы ИИ. Например, активное обучение — это динамическая реконфигурация весов в результате индивидуального обучения робота, а регулярные обновления — это управление процессом
7	Модульный подход	Таблица А.19	Также применимо к элементам технологии ИИ
8	Использование доверительных/проверенных элементов ПО (при наличии)	С.2.10	Также применимо к элементам технологии ИИ. Следует отметить, что верифицированное ПО необходимо не на всех этапах разработки модели ИИ. Оно актуально для выводов, но не для процесса сбора данных
9	Прямая прослеживаемость между спецификацией требований к ПО системы безопасности и архитектурой ПО	С.2.11	Также применимо к элементам технологии ИИ
10	Обратная прослеживаемость между спецификацией требований к ПО системы безопасности и архитектурой ПО	С.2.11	Также применимо к элементам технологии ИИ
11a	Методы структурных диаграмм	С.2.1	Также применимо к элементам технологии ИИ
11b	Полуформальные методы	Таблица А.17	Также применимо к элементам технологии ИИ
11c	Формальные методы проектирования и усовершенствования	В.2.2, С.2.4	Также применимо к элементам технологии ИИ
11d	Автоматическая генерация ПО	С.4.6	Основной принцип разработки ПО подходит и для элементов технологии ИИ
12	Автоматизированные средства разработки спецификаций и проектирования	В.2.4	Также применимо к элементам технологии ИИ
13a	Циклическое поведение с гарантированным максимальным временем цикла	С.3.11	Также применимо к элементам технологии ИИ
13b	Архитектура с временным распределением	С.3.11	Также применимо к элементам технологии ИИ
13c	Управление событиями с гарантированным максимальным временем реакции	С.3.11	Также применимо к элементам технологии ИИ
14	Статическое выделение ресурсов	С.2.6.3	Также применимо к элементам технологии ИИ
15	Статическая синхронизация доступа к разделяемым ресурсам	С.2.6.3	Также применимо к элементам технологии ИИ. Управление может осуществляться с помощью соответствующего встроенного ПО (например, среды выполнения)

Таблица А.3 — Интерпретация проектирования и разработки ПО: инструментальные средства поддержки и языки программирования (*ГОСТ IEC 61508-3—2018,* таблица А.3)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Выбор соответствующего языка программирования	С.4.5	Эти меры применимы для независимых от конкретного использования элементов (например, библиотек CUDA C++) и очень сложны для зависимых от конкретного использования элементов (например, моделей). Другими словами, код, запущенный на объекте, все еще выполняет задачу методов, которые неприменимы для остальной части системы ИИ
2	Строго типизированные языки программирования	С.4.1	
3	Подмножество языка	С.4.2	
4a	Сертифицированные средства и сертифицированные трансляторы	С.4.3	Возможны осложнения, т.к. обычно используется готовое коммерческое ПО (Commercial Off-the-Shelf, COTS). Однако можно также провести различие между обучением и выводом. COTS, такие как TensorFlow, используются для разработки и обучения моделей, но TensorRT преобразует модели в исполняемый движок для выводов, и он сертифицирован
4b	Инструментальные средства, заслуживающие доверия на основании опыта использования	С.4.4	Данный метод представляет высокую важность для развития технологий ИИ

Таблица А.4 — Интерпретация проектирования и разработки ПО: детальное проектирование (*ГОСТ IEC 61508-3—2018*, таблица А.4)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1a	Метод структурированных диаграмм	С.2.1	Также подходит для технологии ИИ, ограниченной аспектами ПО (т. е. независимыми от случая использования элементами) и архитектуры (архитектура модели МО обычно описывается с помощью диаграмм, связей и т. д.). Модульный подход также может быть использован в моделях МО. Скорее, не применим для элементов, связанных с данными
1b	Полуформальные методы	Таблица А.17	
1c	Формальные методы проектирования и усовершенствования	В.2.2, С.2.4	
2	Средства автоматизированного проектирования	В.3.5	
3	Программирование с защитой	С.2.5	
4	Модульный подход	Таблица А.19	
5	Стандарты для проектирования и кодирования	С.2.6, таблица А.11	Стандарты для проектирования применимы и к элементам технологии ИИ. Стандарт программирования (метод «белого ящика») применим для независимых от сценария использования элементов (например, библиотек CUDA C++) и очень сложен для зависимых от сценария использования элементов (например, моделей)
6	Структурное программирование	С.2.7	
7	Использование доверительных/проверенных программных модулей и компонентов (по возможности)	С.2.10	Также применимо к элементам технологии ИИ
8	Прямая прослеживаемость между спецификацией требований к ПО системы безопасности и проектом ПО	С.2.11	Также применимо к элементам технологии ИИ

Таблица А.5 — Интерпретация проектирования и разработки ПО: тестирование и интеграция программных модулей (*ГОСТ IEC 61508-3—2018*, таблица А.5)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Вероятностное тестирование	С.5.1	Также применимо к элементам технологии ИИ. Технология ИИ учится на имеющихся данных: если очевидно, что данные подходят для решения поставленной задачи (по количеству и распределению). Характеристики включают в себя: - определение целевой вероятности; - определение набора тестов, используемых для измерения фактической вероятности; - систематическую спецификацию набора тестов (с целью полноты для достижения желаемой задачи, но также учитывая непреднамеренное поведение)
2	Динамический анализ и тестирование	В.6.5, таблица А.12	Также применимо к элементам технологии ИИ
3	Регистрация и анализ данных	С.5.2	Также применимо к элементам технологии ИИ. Сфера для ИИ: Data Engineering (например, создание, управление, спецификация учебных, валидационных и тестовых наборов данных)
4	Функциональное тестирование и тестирование методом «черного ящика»	В.5.1, В.5.2, таблица А.13	Также применимо к элементам технологии ИИ
5	Тестирование рабочих характеристик	Таблица А.16	Также применимо к элементам технологии ИИ
6	Тестирование, основанное на модели	С.5.27	Также применимо к элементам технологии ИИ
7	Тестирование интерфейса	С.5.3	Также применимо к элементам технологии ИИ
8	Управление тестированием и средства автоматизации	С.4.7	Также применимо к элементам технологии ИИ
9	Прямая прослеживаемость между спецификацией проекта ПО и спецификациями тестирования модуля и интеграции	С.2.11	Также применимо к элементам технологии ИИ
10	Формальная верификация	С.5.12	Некоторый уровень возможен, но практически невозможен для всей системы ИИ

Таблица А.6 — Интерпретация интеграции программируемых электронных устройств (ПО и аппаратные средства) (*ГОСТ IEC 61508-3—2018*, таблица А.6)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Функциональное тестирование и тестирование методом «черного ящика»	В.5.1, В.5.2, таблица А.13	Также применимо к элементам технологии ИИ
2	Тестирование рабочих характеристик	Таблица А.16	Также применимо к элементам технологии ИИ

Окончание таблицы А.6

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
3	Прямая прослеживаемость между требованиями проекта системы и ПО к интеграции программных и аппаратных средств и спецификациями тестирования интеграции программных и аппаратных средств	С.2.11	Также применимо к элементам технологии ИИ

Таблица А.7 — Интерпретация подтверждения соответствия безопасности системы аспектов ПО (*ГОСТ IEC 61508-3—2018*, таблица А.7)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Вероятностное тестирование	С.5.1	См. таблицу А.5, строка 1
2	Моделирование процесса	С.5.18	Также применимо к элементам технологии ИИ
3	Моделирование	Таблица А.15	Также применимо к элементам технологии ИИ
4	Функциональное тестирование и тестирование методом «черного ящика»	В.5.1, В.5.2, таблица А.13	Также применимо к элементам технологии ИИ
5	Прямая прослеживаемость между спецификацией требований к ПО системы безопасности и планом подтверждения соответствия ПО системы безопасности	С.2.11	Также применимо к элементам технологии ИИ
6	Обратная прослеживаемость между планом подтверждения соответствия ПО системы безопасности и спецификацией требований к ПО системы безопасности	С.2.11	Также применимо к элементам технологии ИИ

Таблица А.8 — Интерпретация модификации (*ГОСТ IEC 61508-3—2018*, таблица А.8)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Анализ влияния	С.5.23	Также применимо к элементам технологии ИИ. Дополнение: Анализ влияния рассматривает применимость элемента ИИ в том операционном контексте, в который он интегрирован. Планирование управления изменениями учитывает все прогнозируемые триггерные события, которые могут повлечь за собой изменения, такие как четко запланированные непрерывные изменения, изменения в результате обнаруженных аномалий или изменения в результате устаревания требований. Поскольку изменения могут быть предусмотрены уже во время разработки, управление изменениями явно учитывается уже при планировании безопасности (например, путем определения протокола изменения модели и определения действий, которые должны быть выполнены в этом случае). События, которые могут вызвать изменения, также важны

Окончание таблицы А.8

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
2	Повторная верификация измененных программных модулей	С.5.23	Также применимо к элементам технологии ИИ
3	Повторная верификация программных модулей, на которые оказывают влияние изменения в других модулях	С.5.23	Также применимо к элементам технологии ИИ
4a	Повторное подтверждение соответствия системы в целом	Таблица А.7	Также подходит для элементов технологии ИИ в зависимости от влияния изменений
4b	Регрессионное подтверждение соответствия	С.5.25	Также применимо к элементам технологии ИИ
5	Управление конфигурацией ПО	С.5.24	Также применимо к элементам технологии ИИ
6	Регистрация и анализ данных	С.5.2	Также применимо к элементам технологии ИИ
7	Прямая прослеживаемость между спецификацией требований к ПО системы безопасности и планом модификации ПО (включая повторные верификацию и подтверждение соответствия)	С.2.11	Также применимо к элементам технологии ИИ
8	Обратная прослеживаемость между планом модификации ПО (включая повторную верификацию и подтверждение соответствия) и спецификацией требований к ПО системы безопасности	С.2.11	Также применимо к элементам технологии ИИ

Таблица А.9 — Интерпретация верификации ПО (*ГОСТ IEC 61508-3—2018*, таблица А.9)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Формальное доказательство	С.5.12	Некоторый уровень возможен, но практически невозможен для всего приложения ИИ (из-за размера исполняемого кода формальный анализ работает только для частей кода)
2	Анимация спецификации и тестирования	С.5.26	Также применимо к элементам технологии ИИ
3	Статический анализ	В.6.4, таблица А.18	Эти меры применимы для независимых от конкретного использования элементов (например, библиотек CUDA C++), но могут быть более сложными для зависимых от конкретного использования элементов (например, моделей). Выразительность не такая, как в традиционном коде
4	Динамический анализ и тестирование	В.6.5, таблица А.12	
5	Прямая прослеживаемость между спецификацией проекта ПО и планом верификации (включая верификацию данных) ПО	С.2.11	Также применимо к элементам технологии ИИ
6	Обратная прослеживаемость между планом верификации (включая верификацию данных) ПО и спецификацией проекта ПО	С.2.11	Также применимо к элементам технологии ИИ
7	Численный анализ в автономном режиме	С.2.13	Также применимо к элементам технологии ИИ

Таблица А.10 — Интерпретация оценки функциональной безопасности (*ГОСТ IEC 61508-3—2018*, таблица А.10)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Таблица контрольных проверок	B.2.5	Также применимо к элементам технологии ИИ, рассматриваются специализации ИИ
2	Таблицы решений (таблицы истинности)	C.6.1	Также применимо к элементам технологии ИИ
3	Анализ отказов	Таблица А.14	Также применимо к элементам технологии ИИ
4	Анализ отказов по общей причине различного ПО (если используется различное ПО)	C.6.3	Также применимо к ИИ на системном уровне
5	Структурные схемы надежности	C.6.4	Также применимо к элементам технологии ИИ
6	Прямая прослеживаемость между требованиями раздела 8 и планом оценки функциональной безопасности ПО	C.2.11	Также применимо к элементам технологии ИИ

Таблица А.11 — Интерпретация стандартов для проектирования и кодирования (*ГОСТ IEC 61508-3—2018*, таблица B.1)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Использование стандартов кодирования для сокращения вероятности ошибок	C.2.6.2	Эти меры применимы для элементов, не зависящих от условий использования (например, библиотеки CUDA C++), но очень сложны для элементов, зависящих от условий использования (т. е. моделей). Более того, некоторые из требований *ГОСТ IEC 61508-3* (например, 2, 3a, 3b) иногда не подходят для современной разработки ПО, например объектно-ориентированных языков программирования
2	Не использовать динамические объекты	C.2.6.3	
3a	Не использовать динамические переменные	C.2.6.3	
3b	Проверка создания динамических переменных в неавтономном режиме	C.2.6.4	
4	Ограниченное использование прерываний	C.2.6.5	
5	Ограниченное использование указателей	C.2.6.6	
6	Ограниченное использование рекурсий	C.2.6.7	
7	Не использовать неструктурированное управление в программах, написанных на языках высокого уровня	C.2.6.2	
8	Не использовать автоматическое преобразование типов	C.2.6.2	

Таблица А.12 — Интерпретация динамического анализа и тестирования (*ГОСТ IEC 61508-3—2018*, таблица B.2)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Выполнение тестового примера, связанного с анализом граничных значений	C.5.4	Также применимо к элементам технологии ИИ
2	Выполнение тестового примера, связанного с предполагаемой ошибкой	C.5.5	Также применимо к элементам технологии ИИ
3	Выполнение тестового примера, связанного с введением ошибки	C.5.6	Также применимо к элементам технологии ИИ
4	Выполнение тестового примера, сгенерированного на основе модели	C.5.27	Также применимо к элементам технологии ИИ

Окончание таблицы А.12

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
5	Моделирование реализации	С.5.20	Также применимо к элементам технологии ИИ
6	Разделение входных данных на классы эквивалентности	С.5.7	Также применимо к элементам технологии ИИ
7a	Структурный тест со 100 %-ным охватом (точки входа)	С.5.8	Эти меры применимы как для элементов, независимых от случая использования (например, библиотек CUDA C++), так и для кода, описывающего модель, даже если выразительность не такая, как в традиционном коде. Однако достижение адекватного тестового покрытия входного пространства может представлять сложность
7b	Структурный тест со 100 %-ным охватом (операторы)	С.5.8	
7c	Структурный тест со 100 %-ным охватом (условные переходы)	С.5.8	
7d	Структурный тест со 100 %-ным охватом (составные условия, MC/DC)	С.5.8	

Таблица А.13 — Интерпретация функционального тестирования и проверки методом «черного ящика» (*ГОСТ IEC 61508-3—2018*, таблица В.3)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Выполнение тестового примера на основе причинно-следственных диаграмм	В.6.6.2	Также применимо к элементам технологии ИИ
2	Выполнение тестового примера, сгенерированного на основе модели	С.5.27	Также применимо к элементам технологии ИИ
3	Макетирование/анимация	С.5.17	Также применимо к элементам технологии ИИ
4	Разделение входных данных на классы эквивалентности, включая анализ граничных значений	С.5.7, С.5.4	Также применимо к элементам технологии ИИ
5	Моделирование процесса	С.5.18	Также применимо к элементам технологии ИИ

Таблица А.14 — Интерпретация анализа отказов (*ГОСТ IEC 61508-3—2018*, таблица В.4)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1a	Причинно-следственные диаграммы	В.6.6.2	Также применимо к элементам технологии ИИ. Анализ отказов также учитывает аспекты инжиниринга данных
1b	Анализ методом дерева событий	В.6.6.3	
2	Анализ методом дерева отказов	В.6.6.5	
3	Анализ функциональных отказов ПО	В.6.6.4	

Таблица А.15 — Интерпретация моделирования (*ГОСТ IEC 61508-3—2018*, таблица В.5)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Диаграммы потоков данных	С.2.2	Также применимо к элементам технологии ИИ
2a	Метод конечных автоматов	В.2.3.2	Также применимо к элементам технологии ИИ
2b	Формальные методы	В.2.2, С.2.4	Также применимо к элементам технологии ИИ
2c	Моделирование во времени сетями Петри	В.2.3.3	Также применимо к элементам технологии ИИ
3	Моделирование реализации	С.5.20	Также применимо к элементам технологии ИИ
4	Макетирование/анимация	С.5.17	Также применимо к элементам технологии ИИ
5	Структурные диаграммы	С.2.3	Также применимо к элементам технологии ИИ

Таблица А.16 — Интерпретация тестирования рабочих характеристик (*ГОСТ IEC 61508-3—2018*, таблица В.6)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Проверка на критические нагрузки и стресс-тестирование	С.5.21	Также применимо к элементам технологии ИИ
2	Ограничения на время ответа и объем памяти	С.5.22	Также применимо к элементам технологии ИИ
3	Требования к реализации	С.5.19	Также применимо к элементам технологии ИИ

Таблица А.17 — Интерпретация полуформальных методов (*ГОСТ IEC 61508-3—2018*, таблица В.7)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Логические/функциональные блок-схемы	См. *ГОСТ IEC 61508-3—2018*, таблица В.7, примечание 1	Также применимо к элементам технологии ИИ
2	Диаграммы последовательности действий	См. *ГОСТ IEC 61508-3—2018*, таблица В.7, примечание 1	Также применимо к элементам технологии ИИ
3	Диаграммы потоков данных	С.2.2	Также применимо к элементам технологии ИИ
4a	Конечные автоматы/диаграммы переходов	В.2.3.2	Также применимо к элементам технологии ИИ
4b	Моделирование во времени сетями Петри	В.2.3.3	Также применимо к элементам технологии ИИ
5	Модели данных сущность — связь — атрибут	В.2.4.4	Также применимо к элементам технологии ИИ
6	Диаграммы последовательности сообщений	С.2.14	Также применимо к элементам технологии ИИ
7	Таблицы решений и таблицы истинности	С.6.1	Также применимо к элементам технологии ИИ
8	UML	С.3.12	Также применимо к элементам технологии ИИ

Таблица А.18 — Интерпретация статического анализа (*ГОСТ IEC 61508-3—2018*, таблица В.8)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Анализ граничных значений	С.5.4	Эти меры применимы как для независимых от случая использования элементов (например, библиотек CUDA C++), так и для кода, описывающего модель, даже если выразительность не такая, как в традиционном коде. Однако могут возникнуть сложности с достижением адекватного тестового покрытия входного пространства
2	Таблица контрольных проверок	В.2.5	
3	Анализ потоков управления	С.5.9	
4	Анализ потоков данных	С.5.10	
5	Предположение ошибок	С.5.5	
6a	Формальные проверки, включая конкретные критерии	С.5.14	
6b	Сквозной контроль (ПО)	С.5.15	
7	Тестирование на символьном уровне	С.5.11	
8	Анализ проекта	С.5.16	Также применимо к элементам технологии ИИ

Окончание таблицы А.18

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
9	Статический анализ выполнения программы с ошибкой	B.2.2, C.2.4	Эти меры применимы как для независимых от случая использования элементов (например, библиотек CUDA C++), так и для кода, описывающего модель, даже если выразительность не такая, как в традиционном коде. Однако могут возникнуть сложности с достижением адекватного тестового покрытия входного пространства
10	Временной анализ выполнения при наихудших условиях	C.5.20	Также применимо к элементам технологии ИИ

Таблица А.19 — Интерпретация модульного подхода (*ГОСТ IEC 61508-3—2018*, таблица B.9)

Метод/средство		Ссылка	Интерпретация для элементов технологии систем ИИ
1	Ограничение размера программного модуля	C.2.9	Эти меры применимы для элементов, не зависящих от случая использования (например, библиотеки CUDA C++), а также для кода, описывающего модель, даже если выразительность не такая, как в традиционном коде. Могут возникнуть трудности при тестовом покрытии входного пространства. Что касается размера ПО, то критерием, скорее всего, является не размер, а количество параметров (например, ограниченное количество узлов или соединений или слоев нейронной сети). Сложность также может быть переопределена для МО. Может быть объединена с размером или типами связности между слоями, поскольку глубокие нейронные сети обычно не имеют ветвящихся утверждений
2	Управление сложностью ПО	C.5.13	
3	Ограничение доступа/инкапсуляция информации	C.2.8	
4	Ограниченное число параметров/фиксированное число параметров подпрограммы	C.2.9	
5	Одна точка входа и одна точка выхода в каждой подпрограмме и функции	C.2.9	
6	Полностью определенный интерфейс	C.2.9	Также применимо к элементам технологии ИИ

Приложение В
(справочное)

Примеры применения принципа трехэтапной реализации

В.1 Введение

В настоящем приложении описаны неисчерпывающие примеры применения схемы классификации, описанной в разделе 6, и принципа трехэтапной реализации, описанного в разделе 7.

В.2 Пример использования в автомобильной промышленности

Пример, приведенный в настоящем разделе, представляет собой автомобильную систему, состоящую из двух уровней:

- уровень управления, который отвечает за восприятие окружающей среды, принятие решений, включая планирование маршрутов и управление, в том числе рулевым управлением и торможением;
- уровень защиты, который обеспечивает функции безопасности, такие как определение условий, при которых необходимо выполнить команду защитной остановки или торможения.

Примечание — Уровень управления может быть обозначен как «устройство» по *ГОСТ Р ИСО 26262-1* или «система УО» по *ГОСТ Р МЭК 61508-4*. Уровень защиты может быть обозначен как часть системы, гарантирующей «цель безопасности» по *ГОСТ Р ИСО 26262-1*, или «системы, связанной с безопасностью» по *ГОСТ Р МЭК 61508-4*.

Предполагается, что в систему входят камеры, а соответствующие данные обрабатываются алгоритмом восприятия, основанным на алгоритме глубокого обучения, таким, как глубокие нейронные сети. Примером такого типа глубоких нейронных сетей является DriveNet.

Стандартное изображение этой системы показано на рисунке В.1.

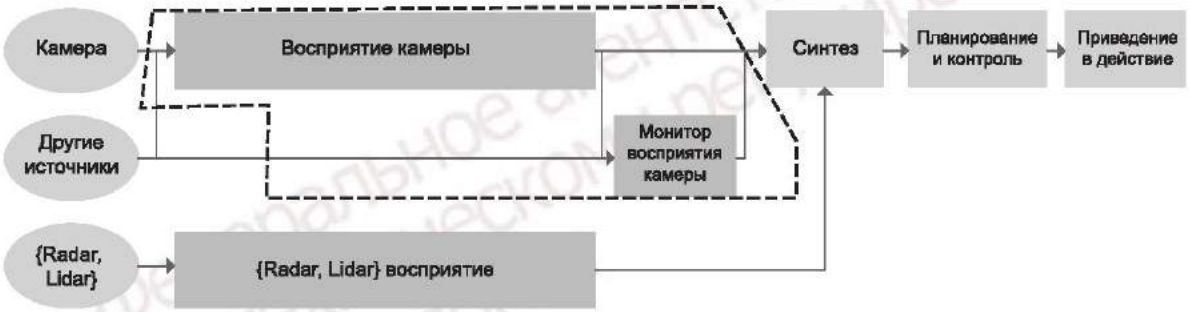


Рисунок В.1 — Пример автомобильной системы

Примечание — На рисунке В.1 светло-серым цветом обозначены входные сигналы датчиков и актуаторы, темно-серым цветом обозначены функции, связанные с восприятием, и соответствующие элементы контроля.

Область применения примера ограничена областью, очерченной пунктирной линией на рисунке В.1, т. е. глубокой нейронной сетью восприятия камеры, связанным с ней каналом восприятия (т. е. камерой) и соответствующими элементами контроля. Другие пути восприятия (lidar, radar), которые могут быть задействованы в системе, соответствующий синтез, а также функции планирования и приведения в действие не входят в эту область.

Технология ИИ, используемая в этой системе, может рассматриваться на уровне использования А1, как описано в 6.2, поскольку она используется в системе Э/Э/ПЭ, имеющей отношение к безопасности, и возможно автоматизированное принятие решений системой ИИ. На основании принципов, описанных в разделе 8, для данного варианта использования можно определить следующие свойства:

- конкретизация: как определить появление пешеходов на изображении?
- интерпретируемость: как получить представление о проектировании?
- генерализация: может ли глубокая нейронная сеть производить интерполяцию во входной области?
- смещение области: работает ли глубокая нейронная сеть в области обучающих данных?
- робастность — безопасность: могут ли небольшие нарушения (вредоносные или нет) изменить выходные данные?
- разнообразие: что означает разнообразие в контексте глубокого обучения и как обеспечить достаточное разнообразие (например, различные архитектуры глубокого обучения, различные наборы данных для обучения)?
- доверие: как рассматривать доверительные уровни в контексте глубокого обучения?

Эти свойства можно сопоставить с тремя этапами принципа реализации, как показано в таблице В.1.

Таблица В.1 — Соотнесение свойств с этапами принципа реализации

Свойства	Сбор данных	Индуцирование знаний из обучающих данных и человеческого знания	Обработка и получение данных на выходе
Конкретизация	—	X	X
Интерпретируемость	—	—	X
Генерализация	—	—	X
Смещение области	—	X	X
Робастность — безопасность	X	—	X
Разнообразие	X	X	X
Доверие	—	—	X

Технология ИИ, используемая в этой системе, может быть отнесена к классу II, поскольку, как показано в таблице В.2, существует возможность определить набор доступных методов и способов, удовлетворяющих свойствам (например, все еще можно использовать определенные компенсационные методы верификации и валидации), так что технология ИИ может соответствовать установленным критериям, а разработка осуществляется в соответствии с подходящими процессами.

В таблице В.2 приведен пример анализа свойств на применимых этапах концепции, а также определены темы, КПЭ, доступные методы и средства для удовлетворения этих свойств.

Таблица В.2 — Пример анализа свойств

Этап: индуцирование знаний из обучающих данных и человеческого знания Желаемое свойство: конкретизация			
Тема	Детали	КПЭ	Доступные методы и источники
Определение набора данных	Количество данных; тип необходимых данных (например, классы объектов, определение данных объекта, погодные условия, географический регион, фоновая сцена); разделение данных между обучением, проверкой и тестированием	Покрытие набора данных; распределение набора данных; пример: набор данных содержит изображения, полученные для различных типов дорог при различных погодных условиях, а сбор данных происходит в дневное время суток	Ручная проверка; активное обучение
Определение политики маркировки	Разметка данных; обработка окклюзии; количество разметчиков, размечающих одни и те же данные	Распределение качества маркировки; пример: границы дорожных полос обозначены пиксель за пикселем. Каждое изображение размечается двумя независимыми разметчиками. 10 % случайно выбранных данных дополнительно размечаются третьим разметчиком	

В.3 Пример использования в робототехнике

Пример, приведенный в настоящем разделе, представляет собой автономную мобильную платформу, которая перевозит материалы по промышленному складу.

Система разделена на прикладную область и область безопасности:

- прикладная область включает беспроводную связь с системой управления автопарком для получения задач и обновлений, а также локальное ПО на устройстве для выполнения задач (локализация, навигация, составление карт);

- область безопасности обеспечивает такие функции безопасности, как аварийная остановка, защитная остановка, ограничение скорости и отключение звука.

Система может быть обозначена как автоматически управляемый промышленный погрузчик — по *[19]* и мобильный промышленный робот (тип B) — по *[20]*.

Сфера применения примера ограничена реализацией функции безопасности защитной остановки, поскольку это единственная функция безопасности, в которой используется МО. Предполагается, что все функции безопасности независимы друг от друга, а прикладная область изолирована от области безопасности.

На рисунке В.2 показано упрощенное представление этой системы, ограниченное компонентами, относящимися к функции безопасности защитной остановки. Датчики камеры вокруг робота передают изображения в нейронную сеть, которая создает изображение глубины. Изображение глубины преобразуется в трехмерное представление сцены. Выполняется проверка, не произошло ли нарушение правил безопасности. Если это так, то двигателю посылается сигнал защитной остановки. Хотя на роботе показаны дополнительные датчики, предполагается, что функция безопасности может быть реализована на каждом датчике независимо (в отличие от «системы систем»).

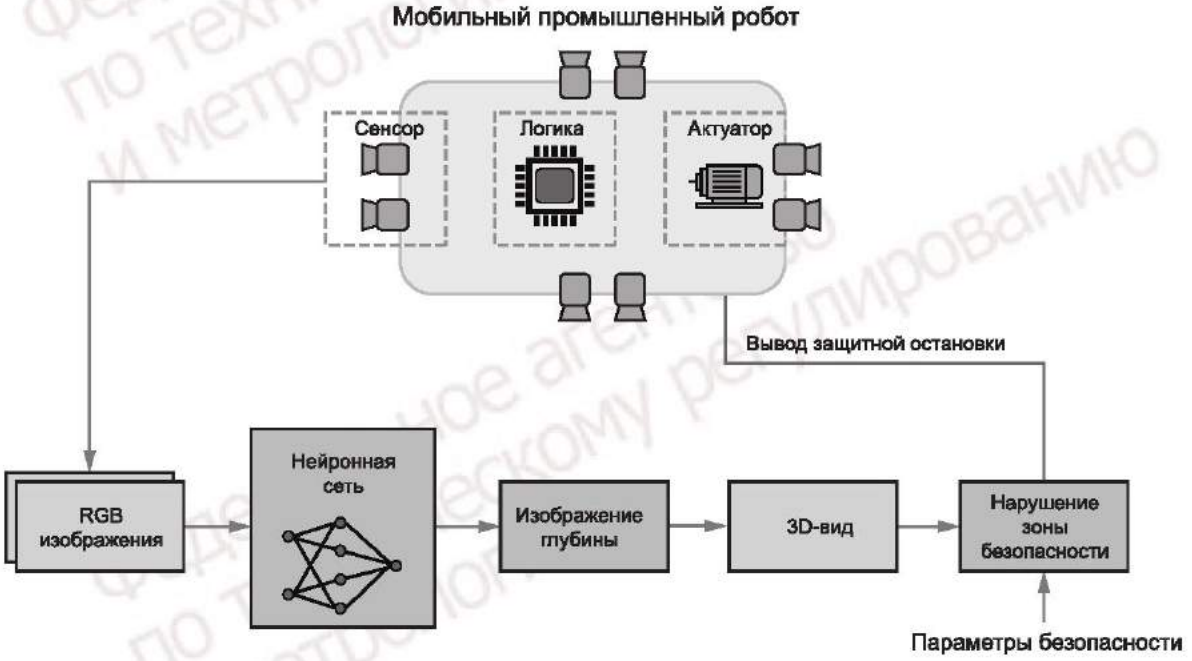


Рисунок В.2 — Пример мобильного промышленного робота

Компоненты ПО темно-серого цвета с жирным черным контуром (т. е. нейронная сеть и изображение глубины) представляют логику и выходные данные, непосредственно создаваемые моделью МО. Все остальные компоненты, выделенные серым цветом, не входят в область применения настоящего стандарта, поскольку они могут быть подтверждены с помощью существующих стандартов. Темно-серые компоненты можно рассматривать как уровень использования А1, как описано в 6.2, поскольку они используются в системе Э/Э/ПЭ, имеющей отношение к безопасности, и возможно автоматизированное принятие решений ИИ.

Исходя из принципов, описанных в разделе 8, следующие свойства могут быть надлежащим образом учтены компонентами ИИ для данной области применения:

- конкретизация. Каковы требования к сети? Как эти требования соотносятся с существующими стандартами для датчиков безопасности, такими как *[18]* и *[17]*? Что представляют собой обучающие изображения для нейронной сети, как эти изображения соотносятся с рабочей средой? Сколько изображений различных классов достаточно для обучения?

- смещение области. Что делать, если среда развертывания отличается от среды, используемой во время обучения?

- проверяемость. Как оценивается производительность нейронной сети? Как эта оценка согласуется с существующими стандартами для датчиков безопасности, такими как *[18]* и *[17]*?

- робастность. Насколько нейронная сеть устойчива к нарушениям входных данных, вызванным различными причинами (аппаратное обеспечение, факторы окружающей среды, изменения в работе, устаревание и т. д.)?

- интерпретируемость. Понятны ли результаты, полученные сетью? Соответствуют ли полученные результаты ожидаемым результатам, определенным требованиями безопасности?

- понятность. Понятны ли компоненты, из которых состоит модель МО? Есть ли причина для выбора вариантов проектирования? Соответствует ли этот выбор входным требованиям?

В таблице В.3 данные свойства соотнесены с трехэтапным принципом реализации из раздела 7.

Таблица В.3 — Соотнесение свойств с этапами принципа реализации

Свойства	Сбор данных	Индуцирование знаний из обучающих данных и человеческого знания	Обработка и получение данных на выходе
Конкретизация	X	X	X
Смещение области	—	X	X
Проверяемость	—	X	X
Робастность	X	—	X
Интерпретируемость	—	X	X
Понятность	—	X	—

В таблице В.4 приведен пример анализа из третьего этапа принципа реализации по отношению к свойству проверяемости.

Таблица В.4 — Пример анализа свойств

Этап: обработка и получение данных на выходе		
Желаемое свойство: проверяемость		
Тема	Детали	Критерии соответствия
Как оценивается производительность нейронной сети?	Для заданного ввода определение того, что является «правильным» выводом сети; определение того, какой диапазон входных данных оценивается	КПЭ на уровне пикселей; КПЭ на уровне изображения; КПЭ на уровне последовательности; КПЭ на уровне набора данных
Как производительность сети соотносится с существующими стандартами и показателями в сфере безопасности?	Сопоставление измеренных критериев производительности сети с существующими стандартными критериями; отслеживание требований от стандартных требований к производительности до сетевых требований	Прослеживаемость требований или документы по составлению карт
Как определить, что процесс проверки является точным (например, неожиданное поведение из-за комбинации факторов, которое невозможно увидеть при тестировании отдельных факторов)?	Одномерное и многомерное тестирование; статистический анализ тестирования случайных переменных; процесс независимой проверки; оценка и/или сертификация средств проверки	Планы тестирования; результаты независимой экспертизы; статистический анализ; квалификация инструментов; FMEA процесса

Окончание таблицы В.4

Этап: обработка и получение данных на выходе		
Желаемое свойство: проверяемость		
Тема	Детали	Критерии соответствия
Как определить момент окончания проверки?	Количество данных для проверки; тип данных проверки и способ их разделения на соответствующие параметры; частота, с которой проводится проверка; частота обновления данных проверки; критерии остановки для проверки	Планы тестирования; предопределенные критерии остановки; FMEA процесса

Аналогичный анализ можно применить ко всем другим свойствам, указанным в таблице В.3, определив набор доступных методов и технологий для удовлетворения свойства. Таким образом, технология ИИ, используемая в данной системе, может считаться классом II, как определено в 6.2.

Приложение С
(справочное)

Возможный процесс и технология для проверки и валидации

С.1 Общие положения

Настоящее приложение дополняет содержание 9.3 и описывает примеры возможных процессов и технических методов.

С.2 Распределение данных, анализ опасностей и оценка риска

Для поддержания достаточного уровня взаимосвязи между распределением данных и анализом опасностей и рисков, описанных в 9.3.2 и 9.3.3 а), могут быть приняты следующие меры.

1) Для требований низкого уровня:

- изучают и записывают основную причину возможного ухудшения безопасности;
- по результатам анализа проектируют данные и отражают их в применимых атрибутах.

2) Для требований среднего уровня, в дополнение к описанным выше мерам для требований низкого уровня:

- анализируют риски ухудшения качества работы системы в целом и его влияние при определенном уровне инженерного охвата и фиксируют результаты в документах;
- анализируют применимость мер к какому-либо из этих рисков, а также связанные с риском атрибуты, содержащиеся во входных данных для компонентов МО;
- анализируют и записывают характерные для приложений характеристики сред, которые способны генерировать входные данные для МО, в зависимости от сложности МО и других аспектов;
- изучают наборы атрибутов и значения атрибутов на основе результатов этого анализа и записывают краткую информацию о таких решениях.

3) Для требований высокого уровня в дополнение к описанным выше мерам для требований низкого и среднего уровня:

- изучают документы результатов прошлых проверок и документы других проверок элементов, которые должны быть извлечены в качестве характеристик системной среды, и записывают краткую информацию об исследованиях, позволяющих извлекать применимые подмножества;
- изучают результаты прошлых исследований в соответствии с областями применения систем и рисками ухудшения качества при использовании общих систем и записывают результаты исследований, включая контекст выбора;
- исключают риски ухудшения качества при использовании общих систем с помощью инженерного анализа, такого как анализ дерева отказов, и записывают их результаты.

Для определения наборов атрибутов, связанных с выявленным риском, примеры аспектов, которые следует учитывать, включают:

- базовые знания о существующем проекте функциональной безопасности, существующих перечнях опасностей в каждой области применения или результатах мозгового штурма на основе примеров опасностей, опасных ситуаций и опасных случаев, приведенных в *ГОСТ ISO 12100*;
- анализ случаев в более ранних системах и аналогичных системах на основе МО;
- пополнение знания предметной области путем мозгового штурма с пользователями;
- пополнение знания о предварительно используемых данных и исключительных случаях, выявленных в ходе испытаний обучения на этапе проверки концепции.

С.3 Покрытие данными для выявленных рисков

Следующим шагом является наличие достаточного количества разнообразных данных по каждому выявленному риску. Проверить распределение значений атрибутов относительно несложно, если определено подмножество данных, относящихся к конкретному риску. Однако, поскольку в типичных случаях использования МО количество атрибутов может стать неуправляемым, проверка всех комбинаций значений атрибутов, вероятно, нереалистична. Вместо этого имеет смысл использовать какой-либо показатель охвата для оценки вероятности того, что все важные из них были рассмотрены. В области исследования тестирования ПО используется несколько показателей тестового покрытия, одним из возможных решений является повторное использование такой концепции для определения покрытия, в частности, концепции комбинаторного тестирования.

Функции объяснимости модели можно использовать для проверки полноты данных в дополнение к проверке на соответствие спецификациям.

Таким образом, некоторые возможные меры для этой подзадачи могут быть установлены следующим образом:

1) Для требований низкого уровня:

- обеспечивают наличие некоторых входных данных для каждого из атрибутов, соответствующих основным факторам риска;

- убеждаются, что некоторые исходные данные соответствуют комбинациям составных факторов риска;
- извлекают атрибуты различий для крайне важных факторов окружающей среды и подготавливают данные, соответствующие комбинациям с серьезными факторами риска.

2) Для требований среднего уровня в дополнение к описанным выше мерам для требований низкого уровня:
- убеждаются, что данные, соответствующие особенно важным факторам риска, в принципе соответствуют стандартам парного охвата.

3) Для требований высокого уровня в дополнение к описанным выше мерам для требований среднего и низкого уровня:
- из инженерных соображений устанавливают стандарты охвата атрибутов и наборы комбинаций значений атрибутов, которые удовлетворяют стандартам охвата;
- определяют требуемый уровень строгости стандартов покрытия (парное покрытие, тройное покрытие и т. д.) с учетом использования системы и степени риска. Стандарты могут быть установлены индивидуально для каждого риска, где это необходимо.

С.4 Разнообразие данных для выявленных рисков

Управление процессом сбора данных позволяет получить разнообразие данных, имеющих схожие значения атрибутов, поскольку в обычном приложении МО мало что известно о таком разнообразии данных. Необходимо тщательно проанализировать источник данных и его обработку, чтобы гарантировать отсутствие предвзятости данных к некоторым конкретным входным условиям.

Некоторые возможные меры для этой подзадачи включают:

1) Для требований низкого уровня:
- анализируют источник и метод получения тестовых наборов данных, чтобы убедиться, что в прикладных ситуациях не обнаружено систематической ошибки;
- исключают выборки без смещения исходных данных для каждого случая, чтобы убедиться, что смещение не обнаружено;
- записывают действия, выполненные для предотвращения предвзятости;
- убеждаются, что наборы обучающих и тестовых данных достаточны для каждого случая, проанализированного на этапе обучения, этапе проверки и т. д.;
- если для какого-либо случая невозможно получить достаточные обучающие данные, пересматривают стандарты охвата и записывают то, что должно быть проверено индивидуально с помощью тестов системной интеграции в соответствии с исходными стандартами.

2) Для требований среднего уровня в дополнение к мерам, описанным выше, для требований низкого уровня:
- оценивают приблизительную вероятность возникновения для каждого значения атрибута или каждого случая;
- проверяют, не отклоняются ли полученные данные от ожидаемого распределения;
- для каждого случая устанавливают разнообразие подмножества данных, соответствующего случаю, используя некоторые числовые показатели или логические критерии. Например, в каждом случае, когда есть какой-либо атрибут, не включенный в этот случай, исключают распределение, связанное с этим атрибутом, и убеждаются в отсутствии значительного смещения.

3) Для требований высокого уровня в дополнение к мерам, описанным выше, для требований среднего и низкого уровней:
- получают показатели охвата данных, включенных в каждый случай. Например, необходимо проверить, существует ли корреляция между данными, отличными от значений атрибутов, включенными в комбинации вариантов, с использованием извлечения признаков или любого другого метода.

См. к 9.3.6 и 9.4 для получения информации об использовании расширения данных и симуляторов для увеличения размера данных.

С.5 Надежность и робастность

Для обеспечения надежности и робастности сгенерированной модели МО необходимо использовать обычные показатели ИИ и набор тестовых данных. Однако обеспечение надежности МО в настоящее время является сложной проблемой.

Информация, представленная в 9.3.3 d), позволяет сделать выводы о возможности применения некоторых методов. Для приложений с более низкими требованиями общих предостережений часто бывает достаточно. Для требований высокого уровня необходимо проведение численного или формального анализа стабильности.

Могут быть использованы некоторые из следующих технологий:

1) Для требований низкого уровня:
- регуляризация;
- кросс-валидация;
- рандомизированное обучение;
- исследование размера модели.

2) Для требований среднего уровня в дополнение к технологиям, описанным выше, для требований низкого уровня:

- состязательное обучение;
- сглаживание;
- генерация состязательных примеров или тесты обнаружения.

3) Для требований высокого уровня в дополнение к технологиям, описанным выше, для требований среднего и низкого уровня:

- оценка максимального безопасного радиуса;
- формально проверенное обучение робастности и сглаживание.

Приложение D
(справочное)

Соответствие между *[1]* и *ГОСТ Р МЭК 61508-1*

[1] описывает жизненный цикл систем ИИ как серию процессов. *ГОСТ Р МЭК 61508-1* описывает жизненный цикл безопасности как серию фаз.

Примечание — *ГОСТ Р МЭК 61508-1* определяет нормативные значения входных данных, деятельности и выходных данных, тогда как *[1]*, основанный на *[21]*, определяет процессы на протяжении всего жизненного цикла в рамках системы.

Оба стандарта определяют набор процессов и фаз на протяжении всего жизненного цикла системы. Оба стандарта позволяют изменять последовательность выполнения каждого процесса или фазы, а также повторять процесс или фазу при условии, что влияние дополнительной работы также учитывается в соответствии со всеми другими процессами или фазами, которые затрагиваются дополнительной работой.

В таблице D.1 приведено сопоставление технических процессов по *[1]* и фаз жизненного цикла по *ГОСТ Р МЭК 61508-1*. Другие процессы по *[1]* также принимаются во внимание, в особенности процесс управления рисками.

Таблица D.1 — Соответствие между *ГОСТ Р МЭК 61508-1* и *[1]*

Жизненный цикл систем безопасности (*ГОСТ Р МЭК 61508-1*)	Жизненный цикл систем ИИ *[1]*. Технические процессы
Концепция (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 1 Концепция)	Процесс анализа бизнеса или задачи; процесс определения требований и потребностей заинтересованных сторон
Определение области применения всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 2 Определение области применения всей системы безопасности)	Процесс определения требований и потребностей заинтересованных сторон; процесс определения системных требований
Анализ опасностей и рисков (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 3 Анализ опасностей и рисков)	Процесс управления рисками
Требования ко всей системе безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 4 Требования ко всей системе безопасности)	Процесс определения требований и потребностей заинтересованных сторон; процесс определения системных требований
Распределение требований по всей системе безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 5 Распределение требований по всей системе безопасности)	Процесс определения архитектуры; процесс определения требований и потребностей заинтересованных сторон; процесс определения системных требований
Планирование эксплуатации и технического обслуживания всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 6 Планирование эксплуатации и технического обслуживания всей системы безопасности)	Процесс эксплуатации; процесс технического обслуживания
Планирование подтверждения соответствия всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 7 Планирование подтверждения соответствия всей системы безопасности)	Процесс подтверждения соответствия
Планирование установки и ввода в эксплуатацию всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 8 Планирование установки и ввода в эксплуатацию всей системы безопасности)	Процесс ввода в действие

Продолжение таблицы D.1

Жизненный цикл систем безопасности (*ГОСТ Р МЭК 61508-1*)	Жизненный цикл систем ИИ *[1]*. Технические процессы
Спецификация требований к Э/Э/ПЭ системе безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 9 Спецификация требований к Э/Э/ПЭ системе безопасности)	Процесс определения требований и потребностей заинтересованных сторон; процесс определения системных требований
Установка и ввод в эксплуатацию всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 12 Установка и ввод в эксплуатацию всей системы безопасности)	Процесс ввода в действие
Подтверждение соответствия всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 13 Подтверждение соответствия всей системы безопасности)	Процесс подтверждения соответствия
Спецификация требований проектирования Э/Э/ПЭ системы (*ГОСТ Р МЭК 61508-1—2012*, рисунок 3: 10.1 Спецификация требований проектирования Э/Э/ПЭ системы)	Процесс определения архитектуры; процесс определения проектирования; процесс системного анализа
Планирование подтверждения соответствия безопасности Э/Э/ПЭ системы (*ГОСТ Р МЭК 61508-1—2012*, рисунок 3: 10.2 Планирование подтверждения соответствия безопасности Э/Э/ПЭ системы)	Процесс подтверждения соответствия
Проектирование и разработка системы (*ГОСТ Р МЭК 61508-1—2012*, рисунок 3: 10.3 Проектирование и разработка Э/Э/ПЭ системы, включая СИС и ПО)	Процесс системного анализа; процесс разработки
Интеграция системы (*ГОСТ Р МЭК 61508-1—2012*, рисунок 3: 10.4 Интеграция Э/Э/ПЭ системы)	Процесс интеграции
Спецификация требований к ПО системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 4: 10.1 Спецификация требований к ПО системы безопасности)	Процесс определения требований и потребностей заинтересованных сторон; процесс системного анализа
Проектирование и разработка ПО (*ГОСТ Р МЭК 61508-1—2012*, рисунок 4: 10.3 Проектирование и разработка ПО)	Процесс сбора данных; процесс инжиниринга данных ИИ; процесс разработки
План подтверждения соответствия для аспектов ПО системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 4: 10.2 План подтверждения соответствия для аспектов ПО системы безопасности)	Процесс подтверждения; процесс подтверждения соответствия
Интеграция программируемого оборудования (технические средства и ПО) [*ГОСТ Р МЭК 61508-1—2012*, рисунок 4: 10.4 Интеграция программируемого оборудования (технические средства и ПО)]	Процесс интеграции
Процедуры эксплуатации и технического обслуживания ПО (*ГОСТ Р МЭК 61508-1—2012*, рисунок 4: 10.5 Процедуры эксплуатации и технического обслуживания ПО)	Процесс эксплуатации; процесс технического обслуживания
Подтверждение соответствия аспектов ПО системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 4: 10.6 Подтверждение соответствия аспектов ПО системы безопасности)	Процесс подтверждения соответствия

Окончание таблицы D.1

Жизненный цикл систем безопасности (*ГОСТ Р МЭК 61508-1*)	Жизненный цикл систем ИИ *[1]*. Технические процессы
Процедуры установки, ввода в действие, эксплуатации и технического обслуживания Э/Э/ПЭ системы (*ГОСТ Р МЭК 61508-1—2012*, рисунок 3: 10.5 Процедуры установки, ввода в действие, эксплуатации и технического обслуживания Э/Э/ПЭ системы)	Процесс ввода в действие
Подтверждение соответствия безопасности Э/Э/ПЭ системы (*ГОСТ Р МЭК 61508-1—2012*, рисунок 3: 10.6 Подтверждение соответствия безопасности Э/Э/ПЭ системы)	Процесс подтверждения соответствия; постоянный процесс подтверждения соответствия
Эксплуатация, техническое обслуживание и ремонт всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 14 Эксплуатация, техническое обслуживание и ремонт всей системы безопасности)	Процесс эксплуатации; процесс технического обслуживания
Реконструкция и модификация всей системы безопасности (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 15 Реконструкция и модификация всей системы безопасности)	Процесс технического обслуживания
Вывод из эксплуатации или утилизация (*ГОСТ Р МЭК 61508-1—2012*, рисунок 2: 16 Вывод из эксплуатации или утилизация)	Процесс утилизации

Приложение ДА
(справочное)

Текст невключенной таблицы 3 примененного проекта международного стандарта

Таблица 3 — Примеры элементов, включенных в процесс создания и выполнения модели для МО

Элемент технологии (на примере МО)	Примеры (неполный список)[b]
Граф приложения	Graph eXchange Format (GXF), graph in YAML Ain't Markup Language (YAML), recently qualified teacher (rqt) graph in Robot Operating System (ROS/ROS2)
Фреймворк МО[a]	TensorFlow, PyTorch, Keras, mxnet, Microsoft Cognitive Toolkit (CNTK), Caffe, Theano
Язык модели МО	Open Neural Network Exchange (ONNX), Neural Network Exchange Format (NNEF)
Компилятор графов МО	TensorRT, GLOW, Multi-Level Intermediate Representation (MLIR), nGraph, Tensor Virtual Machine (TVM), PlaidML, Accelerated Linear Algebra (XLA)
Компилятор машинного кода	Gcc, nvcc, clang/llvm, SYCL, dpc++, OpenCL, openVX
Библиотеки	CUDA C++, XMMA/SASS kernels, NumPy, SciPy, Pandas, Matplotlib, CUDA Deep Neural Network (cuDNN), SYCL DNN, oneAPI Deep Neural Network (OneDNN), Math Kernel Library (MKL)
Программа в машинном коде	Исполняемый машинный код, скомпилированный для следующих архитектур: aarch64, PTX, RISC-V, AMD64, x86/64, PowerPC
Аппаратные средства	GPU, CPU, FPGA, ASIC, accelerators

[a] Платформа для МО включает инструменты, библиотеки и ресурсы сообщества.

[b] Торговые марки приведены в интересах пользователей и безопасности. Это не подразумевает поддержки и одобрения этих продуктов со стороны ИСО или МЭК.

Примечание — В этой таблице не различаются элементы для обучения и для логического вывода.

Библиография

[1]	ИСО/МЭК FDIS 5338:2023	Информационные технологии. Искусственный интеллект. Процессы жизненного цикла систем искусственного интеллекта (Information technology — Artificial intelligence — AI system life cycle processes)
[2]	ИСО/МЭК 22989:2022	Информационная технология. Искусственный интеллект. Концепции и терминология искусственного интеллекта (Information technology — Artificial intelligence — Artificial intelligence concepts and terminology)
[3]	ISO/IEC/IEEE 29119-1:2022	Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Основные понятия (Software and systems engineering — Software testing — Part 1: General concepts)
[4]	ИСО 21448:2022	Дорожные транспортные средства. Безопасность заданных функций (Road vehicles — Safety of the intended functionality)
[5]	*ISO/AWI TS 5083*	*Дорожные транспортные средства. Безопасность автоматизированных систем вождения. Проектирование, проверка и валидация (Road vehicles — Safety for automated driving systems — Design, verification and validation)*
[6]	ISO/AWI PAS 8800	Дорожные транспортные средства. Безопасность и искусственный интеллект (Road vehicles — Safety and artificial intelligence)
[7]	ИСО 13849-1:2015	Безопасность оборудования. Элементы систем управления, связанные с безопасностью. Часть 1. Общие принципы конструирования (Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design)
[8]	ИСО 13849-2:2012	Безопасность машин. Детали систем управления, связанные с обеспечением безопасности. Часть 2. Валидация (Safety of machinery — Safety-related parts of control systems — Part 2: Validation)
[9]	ИСО 26262-11:2018	Дорожные транспортные средства. Функциональная безопасность. Часть 11. Руководящие указания по применению ISO 26262 к полупроводникам (Road vehicles — Functional safety — Part 11: Guidelines on application of ISO 26262 to semiconductors)
[10]	МЭК 62061:2021	Безопасность оборудования. Функциональная безопасность систем управления, связанных с безопасностью (Safety of machinery — Functional safety of safety-related control systems)
[11]	ISO/IEC TS 6254	Информационные технологии. Искусственный интеллект. Подходы и методы объяснимости моделей машинного обучения и систем искусственного интеллекта (Information technology — Artificial intelligence — Objectives and methods for explainability of ML models and AI systems)
[12]	ISO/IEC TR 29119-11:2020	Системная и программная инженерия. Тестирование программного обеспечения. Часть 11. Руководящие указания по тестированию систем искусственного интеллекта (AI) (Software and systems engineering — Software testing — Part 11: Guidelines on the testing of AI-based systems)
[13]	ИСО/МЭК 5259 (все части)	Искусственный интеллект. Качество данных для аналитики и машинного обучения (Artificial intelligence — Data quality for analytics and machine learning)
[14]	*ISO/IEC/IEEE 29119-4:2021*	*Системная и программная инженерия. Тестирование программного обеспечения. Часть 4. Методы тестирования (Software and systems engineering — Software testing — Part 4: Test techniques)*
[15]	BS EN 50128:2011+A2:2020	Приложения для железных дорог. Системы связи, сигнализации и обработки данных. Программное обеспечение для систем управления и защиты железных дорог (Railway applications. Communication, signalling and processing systems. Software for railway control and protection systems)
[16]	ISO/IEC/IEEE 12207:2017	Системная и программная инженерия. Процессы жизненного цикла программных средств (Systems and software engineering — Software life cycle processes)

[17]	IEC TS 62998-1:2019	Безопасность механизмов. Датчики, связанные с безопасностью, используемые для защиты людей (Safety of machinery — Safety-related sensors used for the protection of persons)
[18]	МЭК 61496 (все части)	Безопасность механизмов. Защитная электрочувствительная аппаратура (Safety of machinery — Electro-sensitive protective equipment)
[19]	ИСО 3691-4:2020	Автопогрузчики промышленные. Требования безопасности и верификация. Часть 4. Автоматически управляемые промышленные погрузчики и их системы (Industrial trucks — Safety requirements and verification — Part 4: Driverless industrial trucks and their systems)
[20]	ANSI/RIA R15.08-1-2020	Промышленные мобильные роботы. Требования безопасности. Часть 1. Требования к промышленному мобильному роботу (Industrial Mobile Robots — Safety Requirements — Part 1: Requirements For The Industrial Mobile Robot)
[21]	ISO/IEC/IEEE 15288:2023	Системная и программная инженерия. Процессы жизненного цикла систем (Systems and software engineering — System life cycle processes)

УДК 004.01:006.354 ОКС 35.020

Ключевые слова: искусственный интеллект, функциональная безопасность, системы искусственного интеллекта

Редактор *Л.В. Коретникова*
Технический редактор *И.Е. Черепкова*
Корректор *Л.С. Лысенко*
Компьютерная верстка *М.В. Малеевой*

Сдано в набор 30.11.2023. Подписано в печать 13.12.2023. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 7,90. Уч.-изд. л. 6,72.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru info@gostinfo.ru