На правах рукописи

Панков Илья Анатольевич

МОДЕЛИ И АЛГОРИТМЫ ВЫЯВЛЕНИЯ ДЕФЕКТОВ В УСТРОЙСТВАХ СВЯЗИ РАСПРЕДЕЛЕННОГО ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА С ПРИМЕНЕНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ

Специальность 2.3.1. Системный анализ,
управление и обработка информации, статистика

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Омск – 2026

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Омский государственный технический университет» (ОмГТУ).

Научный руководитель: **Денисова Людмила Альбертовна,**
доктор технических наук, профессор.

Официальные оппоненты: **Петренко Сергей Анатольевич,**
доктор технических наук, профессор, руководитель группы научного центра информационных технологий и искусственного интеллекта автономной некоммерческой образовательной организации высшего образования «Научно-технологический университет «Сириус».
Егоров Александр Алексеевич,
кандидат технических наук, руководитель разработки по направлению дополнительного программного обеспечения группы компаний общества с ограниченной ответственностью «Цифра».

Ведущая организация: Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», г. Екатеринбург.

Защита состоится «14» апреля 2026 г. в 15:00 часов на заседании диссертационного совета 24.2.350.10, созданного на базе ОмГТУ, по адресу: 644050, Омск, просп. Мира, д. 11, ауд. П-202.

С диссертацией можно ознакомиться в библиотеке ОмГТУ и на сайте www.omgtu.ru.

Автореферат разослан «__ ___» _____ 2026 года.

Ученый секретарь
диссертационного совета 24.2.350.10,
кандидат технических наук, доцент А.С. Грицай

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. Переход промышленности в сегменте критической информационной инфраструктуры на отечественные распределенные программно-аппаратные комплексы (ПАК) в соответствии с указом Президента РФ № 166 «О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации» требует выполнения ряда стандартов в области качества и надежности. Дефекты программного обеспечения (ПО), аппаратные неисправности выявляются с помощью методики имитации неисправностей. Также необходимо обеспечить выявление программных уязвимостей, которые могут привести к атакам на распределенные ПАК. Включение дополнительных устройств и подсистем для выполнения поставленных задач требует дополнительных методов тестирования не только в процессе разработки, но и в процессе эксплуатации, поскольку дефекты, не выявленные ранее, могут проявится при изменении состава устройств ПАК из-за свойства эмерджентности.

Современные распределенные ПАК обладают рядом существенных проблем, обусловленных сложностью архитектуры и повышенными требованиями к устойчивости к отказам и сбоям. К примеру, для доверенных ПАК ГОСТ 56939-2024 «Разработка безопасного программного обеспечения» определяет необходимость тестирования с помощью фаззинга. Методика фаззинга - тестирование ПО, заключающаяся в подаче в программу случайных или специально сформированных входных данных (как корректных, так и некорректных) для выявления ошибок, уязвимостей и нештатного поведения, мутированных данных, не рассмотрена в стандарте, что требует от разработчиков ПАК выбирать методы и средства проведения тестирования и получения количественных оценок для комплекса, эффективность которых до сих пор является вопросом для исследователей ПО.

Актуальной задачей является конфигурация систем имитации неисправностей не для выявления дефектов отдельного устройства, а совокупности устройств контролируемых подсистем. В распределенных ПАК для имитации неисправностей не всегда доступен необходимый набор

средств, учитывающий реконфигурацию подсистем управления. Отдельного внимания требует проблема интеграции гетерогенных устройств с различной архитектурой, включая встраиваемые системы, помимо архитектуры x86. В таких системах необходимо проведение имитации неисправностей для выявления потенциальных отказов и сбоев, возникающих в результате загрузки ПО в реальную аппаратную среду, т.е. контроль проявления свойства эмерджентности. Таким образом, разработка и эксплуатация современных ПАК требует комплексного подхода, направленного на повышение надежности, безопасности и устойчивости к возможным отказам. Для устройств подсистем ПАК также существует необходимость улучшения точности и времени проведения имитация неисправностей согласно с международными и отечественными нормативами: FMECA, FMEDA и ГОСТ Р 58412-2019. Таким образом, архитектура распределенных ПАК должна содержать механизмы для выявления неисправностей и в отдельных устройствах, и в целых подсистемах. Сбой одного компонента может вызвать лавинообразный отказ целых подсистем, особенно с учетом работы в недоверенной среде, где возможны целенаправленные атаки на проектируемые системы. Дополнительные сложности возникают в области восстановления данных при сбоях и отказах распределенных систем, что требует разработки и внедрения надежных алгоритмов уведомления о событиях системы. Важную роль играет построение таких алгоритмов и архитектур, для которых можно обеспечить выполнение современных стандартов при разработке и эксплуатации в реальные сроки для обеспечения стратегических задач промышленности в сжатые сроки. Решение этих задач во всем многообразии на данный момент ограничено в условиях экономической целесообразности выделения ресурсов на техническое проектирование систем. Поэтому применение системного анализа позволит спроектировать архитектуру системы, которая будет включать набор алгоритмов и методов для минимизации рисков возникновения отказов и сбоев, а также ускорение процессов выявления дефектов в условиях возможных атак.

Состояние вопроса. Вопросами проектирования отказоустойчивых программно-аппаратных систем занимались ученые Avizienis A., Rannels D., Williams R., Giovanni De Micheli, Marilyn Wolf, Rolf Ernst, Leslie Lamport, Липаев В.В., Полетаев И.В. Вопросами проектирования средств имитации

неисправностей для устройств и комплексов занимались ученые Avizienis A., Савельев А.Ю., Лучинин В.В., Чекмарев С.А, Похабов Ю.П. Вопросами повышения качества и надежности ПО посвящены работы ученых: Царев В.М., Ковалев И.В., Boehm B. W., Meyer J., Lyu M. R., Levendel Y., Shooman M. L., Tai A., Xie M., Zhou L. Исследованиями свойств ПО и безопасности программ посвящены работы: Berman O., Choi J. G, Epstein D., She D., Pei K. Вопросами кибербезопасности, моделирования угроз и устойчивости к атакам в IoT занимаются ученые Stolfo S., Jha S., Zamboni D., Kruegel C., Vigna G., Антонов С.Г.

Основная идея работы заключается в том, что повышение объема выявляемых дефектов распределенного ПАК можно обеспечить за счет их предварительной классификации путем совместного применения сверточной нейросети (обнаружение отказов) и нечеткого логического вывода (обнаружение ИТВ), а также за счет разработки алгоритма тестирования с помощью имитации искажения данных, направленной на потенциально уязвимые места в ПО, выявленные нейросетевой большой языковой моделью на основе анализа текстов программной документации.

Целью диссертационной работы является повышения объема выявляемых дефектов в ПАК путем их предварительной классификации с применением нейросетевых технологий, а также автоматизации тестирования ПО за счет направленных имитационных воздействий на потенциально уязвимые места программы. Для достижения указанной цели в работе поставлены и решены следующие **задачи:**

1. Провести анализ проблемы проявления новых свойств при реконфигурации, внешних воздействиях и возникновении отказов и сбоев подсистем ПАК с помощью технологий тестирования (FMEA, FMECA, FMEDA, Fault Injection) для выявления отказов и сбоев, а также ускорения их обнаружения средствами ПАК.
2. Разработать модель и алгоритм для определения видов дефектов на основе временных и спектральных характеристик акустических сигналов, формируемых средствами ПАК в условиях информационно-технических воздействий (ИТВ), и возможности разграничения аппаратного и техногенного дефектов.

3. Разработать алгоритм имитации неисправностей с применением техники фаззинга для повышения объема выявленных программных дефектов и сокращения времени тестирования распределенных ПАК в процессе их разработки, тестирования и эксплуатации.

4. Разработать модель и алгоритм выявления критических уязвимостей в ПО распределенного ПАК с помощью нейросетевой большой языковой модели (LLM) для совместного использования с техникой направленного фаззинга на потенциальные уязвимости при тестировании ПАК, позволяющего повысить объем выявляемых ошибок в ПО.

5. Создать программный комплекс для тестирования ПО распределенного ПАК и провести экспериментальные исследования, подтверждающие повышение объема выявляемых дефектов при использовании предлагаемых моделей и алгоритмов. Внедрить полученные результаты в виде алгоритмов и программ, применяемых для подсистем распределенного комплекса.

Научная новизна. В процессе исследований получены следующие новые научные результаты.

1. Разработаны модель и алгоритм для определения вида дефекта на основе сверточной нейронной сети, обрабатывающей временные и спектральные характеристики акустических сигналов с датчиков распределенного ПАК. Основным преимуществом алгоритма, в отличие от существующих, является возможность оценить степень уверенности в отсутствии информационно-технических воздействий на ПАК и при высокой степени уверенности в отсутствии ИТВ разграничить аппаратный и техногенный дефекты.

2. Разработаны модель и алгоритм для выявления информационно-технического воздействия на распределенный ПАК, основанные на использовании нечеткого логического вывода. Отличительной особенностью алгоритма является возможность в условиях действия кибератак на данные разграничить виды дефектов (аппаратный дефект или техногенное воздействие).

3. Разработан модифицированный алгоритм фаззинга для выявления ошибок в ПО устройств связи ПАК, отличительной особенностью которого является совместное применение техники фаззинга для имитации неисправностей и LLM для анализа текстов программной технической документации, позволяющее за счет направленного фаззинга на

потенциально уязвимые места программы повысить число выявленных ошибок по сравнению с традиционным алгоритмом фаззинга.

Практическая значимость работы заключается в разработке:

– архитектуры анализа проблемных ситуаций распределенного ПАК с множественным доступом, который реализует тестирование подсистем с помощью имитации неисправностей случайного внесения искажений в данные на различных уровнях модели OSI и выявляет системные дефекты на основе обработки информации, полученной и обработанной в центре управления.

– алгоритма работы распределенного ПАК, позволяющего выявить основные дефекты и уязвимости подсистем путем сравнения параметров его состояния с известными или ожидаемыми значениями, результатом которого является выявление точек возникновения отказов и сбоев с применением системы нечеткой логики, статистического анализа.

Внедрение результатов исследований. Разработанные методики, алгоритмы и программно-алгоритмические средства для имитации и выявления неисправностей проверяемых устройств внедрены в ООО «СОФТЭНК».

Объектом исследования является распределенный ПАК с набором устройств, имеющих различную программно-аппаратную базу, которые могут содержать набор дефектов и программных уязвимостей.

Предметом исследования являются математические модели, методы и алгоритмы, предназначенные для обеспечения устойчивости распределенных программно-аппаратных комплексов к отказам и сбоям.

Методология исследования базируется на основах системного анализа, методах теории вероятностей и математической статистике; теории принятия решений; методов тестирования и разработки программно-аппаратных систем, методике имитации неисправностей «*fault injection*» и технике фаззинга, машинном обучении.

Основные положения, выносимые на защиту:

1. Модели и алгоритмы классификации дефектов распределенного ПАК, отличительной особенностью которых является совместное применение сверточной нейросети (для обнаружения и локализации отказов) и нечеткого логического вывода (для выявления информационно-технических

воздействий – кибератак). Предлагаемая гибридная модель позволяет выполнять функции по распознаванию и разграничению аппаратных и техногенных дефектов, обеспечивая высокую точность классификации.

2. Модели и алгоритмы тестирования программного обеспечения, распределенного ПАК, отличительной особенностью которых является совместное применение техники фаззинга (для имитации неисправностей случайным искажением входных данных) и нейросетевой большой языковой модели LLM (для выявления критических уязвимостей в ПО). Предлагаемый гибридный алгоритм позволяет увеличить покрытие программного кода при уменьшении временных затрат на проведение тестирования за счёт направленного фаззинга для наиболее уязвимых мест программы.

3. Проблемно-ориентированный программный комплекс, реализующий выявление дефектов в устройствах связи распределенного ПАК. Результаты экспериментальных исследований для разграничения аппаратных и техногенных отказов и сбоев, в том числе при низкой степени уверенности в отсутствии информационно-технических воздействий, а также ошибок ПО.

Соответствие паспорту специальности. Диссертация соответствует областям исследований: п. 4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации искусственного интеллекта», п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 11 «Методы и алгоритмы прогнозирования и оценки эффективности, качества и надежности сложных систем».

Достоверность полученных результатов работы подтверждена экспериментальными исследованиями и основана на апробированных научных положениях и методах исследования, корректном применении математического аппарата, согласованности новых результатов с известными теоретическими положениями. Обоснованность и достоверность результатов диссертации подтверждается результатами экспериментальной проверки разработанных моделей и алгоритмов.

Апробация результатов исследования. Результаты работы представлялись на следующих конференциях: Региональная молодежная научно-практическая конференция «Нанотехнологии. Информация.

Радиотехника» (г. Омск, 2022); XV Межрегиональной научно-практической конференции «Приборостроение и информационные технологии» (г. Омск, 2022); VII Международная научно-практическая конференция «Мехатроника, автоматика и робототехника» (г. Санкт-Петербург, 2023); XIV Всероссийская научно-практическая конференция «Информационные технологии и автоматизация управления» (г. Омск, 2023).

Публикации по теме исследования. По результатам исследований опубликовано 11 научных работ, в том числе 6 научных статей в рецензируемых научных изданиях, рекомендованных ВАК при Минобрнауки России, 2 свидетельства о государственной регистрации программ для ЭВМ.

Личный вклад автора. Решение задач диссертации, разработанные алгоритмы, их программная реализация, экспериментальные и теоретические результаты, представленные в работе и выносимые на защиту, разработаны лично автором.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, изложенных на 146 страницах машинописного текста, содержит 46 рисунков, 27 таблиц, список использованных источников из 134 наименований, 2 приложения на 4 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, проведен анализ степени разработанности исследуемой научной проблемы и обоснованы подходы к ее решению, приведены цель и задачи диссертационного исследования; научная новизна; практическая значимость работы; методы исследования; положения, выносимые на защиту; степень достоверности и апробация полученных результатов.

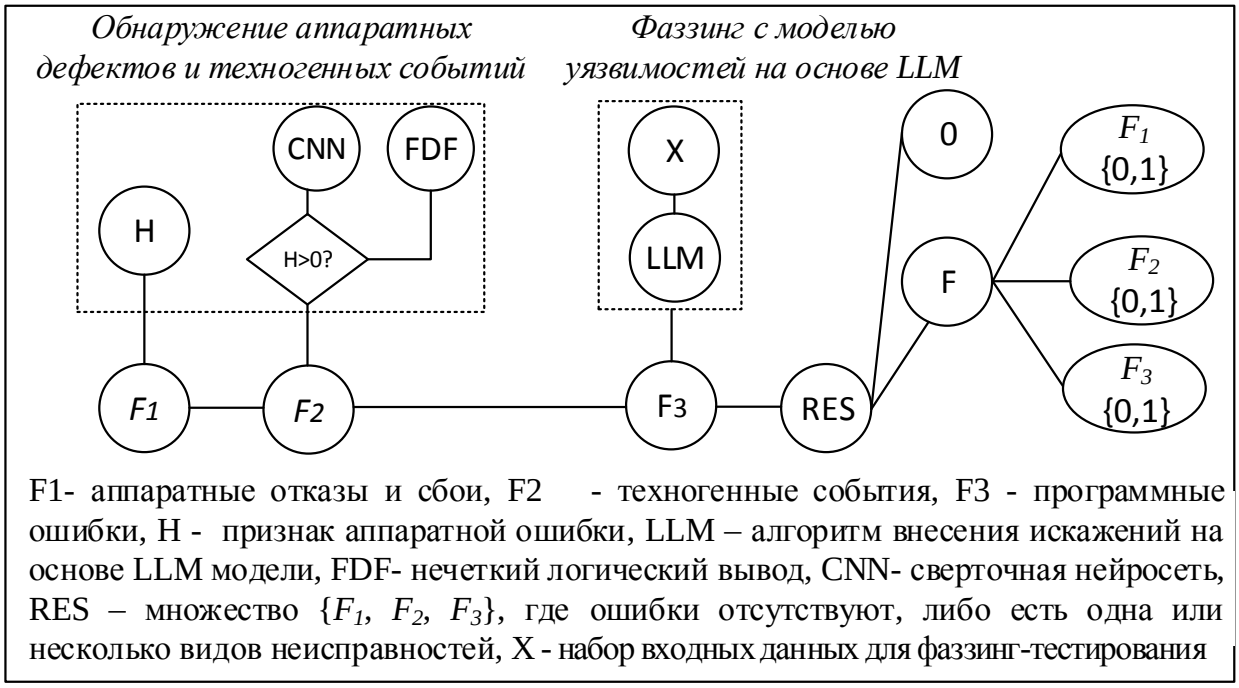
Первая глава посвящена анализу современного состояния проблемы устойчивости к отказам и сбоям распределенных ПАК. Проблема зафиксирована в международных методиках FMEDA (failure modes, effects, and diagnostic analysis) и FMECA (failure mode, effects, and criticality analysis).

Отказы и сбои распределенных ПАК рассматриваются как проявление свойства эмерджентности, которое приводит к нежелательным последствиям, которые невозможно учесть на этапе проектирования.

В связи с тем, что используемые подсистемы комплекса могут содержать отказы и сбои, а также уязвимости, которые могут быть использованы для проведения целенаправленных атак на инфраструктуру, то основной проблемой является как поиск точек возникновения отказов и сбоев, обнаружения дефектов ПО, техногенных событий с учетом необходимости адаптивности комплекса к новой конфигурации и выполнения требований стандартов.

Во второй главе представлена методика анализа проблемных ситуаций в распределенных ПАК и обнаружения проявлений на этапе проектирования. Постоянный мониторинг необходим из-за изменения структуры системы при добавлении или удалении устройств. Высокая связность компонентов усложняет разделение программных и аппаратных отказов, поэтому требуются дополнительные инструменты для определения их природы - программной, аппаратной или вызванной внешними воздействиями. Для решения этой задачи используется комбинация алгоритмов имитации неисправностей на базе нейронных сетей и нечеткой логики, что позволяет уточнять причины отказов и сбоев. Если происхождение отказа или сбоя аппаратное H, то программное выявление ошибок избыточно, необходимо ускоренно выявить и принять меры для его устранения. В ряде случаев определение причины отказа или сбоя, которая может быть, как программной, так и аппаратной, является сложной задачей. Также часть алгоритмов диагностики работает не корректно в условиях, если идет целенаправленная атака S на узлы или подсистемы, поэтому предложена программная структура комплекса выявления неисправностей ПАК.

Работа распределенного программно-аппаратного комплекса строится по многоуровневой схеме, где каждый этап последовательно уточняет и дополняет результаты предыдущего. Такая структура обеспечивает комплексное тестирование - от анализа программного кода и алгоритмических уязвимостей до аппаратных, сетевых и сигнатурных отклонений в работе системы (рисунок 1).



F1- аппаратные отказы и сбои, F2 - техногенные события, F3 - программные ошибки, H - признак аппаратной ошибки, LLM – алгоритм внесения искажений на основе LLM модели, FDF- нечеткий логический вывод, CNN- сверточная нейросеть, RES – множество {F_1, F_2, F_3}, где ошибки отсутствуют, либо есть одна или несколько видов неисправностей, X - набор входных данных для фаззинг-тестирования

Рисунок 1 – Программная структура комплекса выявления неисправностей ПАК

Модуль обнаружения дефектов предназначен для получения признака атак *S* и проверки на отсутствие отказов оборудования с помощью датчиков устройств. Далее идет анализ на техногенные события. Для этого применяется анализ по сети на наличие признака атак на данные, позволяющие переключиться на более устойчивый алгоритм FDF (Fuzzy Detection Fault или нечеткий логический вывод): модуль нечеткой логики предназначен для получения признака техногенных событий за счет выявления аномалий в данных сигналов на базе CNN (Convolutional Neural Networks) или FDF.

Далее проводится выявление ошибок кода: на этом этапе применяется модифицированный фаззинг, в котором генерация тестовых воздействий осуществляется на основе обученной модели уязвимостей, созданной с использованием LLM. Эта модель учитывает известные шаблоны функций и интерфейсов, извлеченные из технической документации на устройства, а также данные об их типичных параметрах и режимах работы. Такой подход позволяет интегрировать генерацию аномальных входных данных непосредственно в процесс тестирования, а также за счёт интеграции контекста ранее обнаруженных в процессе тестирования дефектов, выявлять уязвимости и формировать направленные на конкретные алгоритмы работы устройств и подсистем информационно-технические воздействия, расширяя объем исследованного кода для фаззинга в рамках единого цикла тестирования.

Третья глава посвящена разработке и исследованию алгоритмов проведения испытаний ПАК для разграничения причин возникновения отказов и сбоев. Предлагаемые решения ориентированы на формирование интеллектуальной системы тестирования, способной в автоматизированном режиме выявлять техногенные события, определять причины нарушений и обеспечивать приоритетную реакцию на критические дефекты.

Поскольку внешние воздействия могут включать атаки на подсистемы, то предлагается использование коэффициента степени уверенности (DR, %) для переключения на алгоритм на базе нечеткой логики при наличии атак в случаях, когда необходимо принять дополнительное решение по составу внешних факторов (рисунок 2).

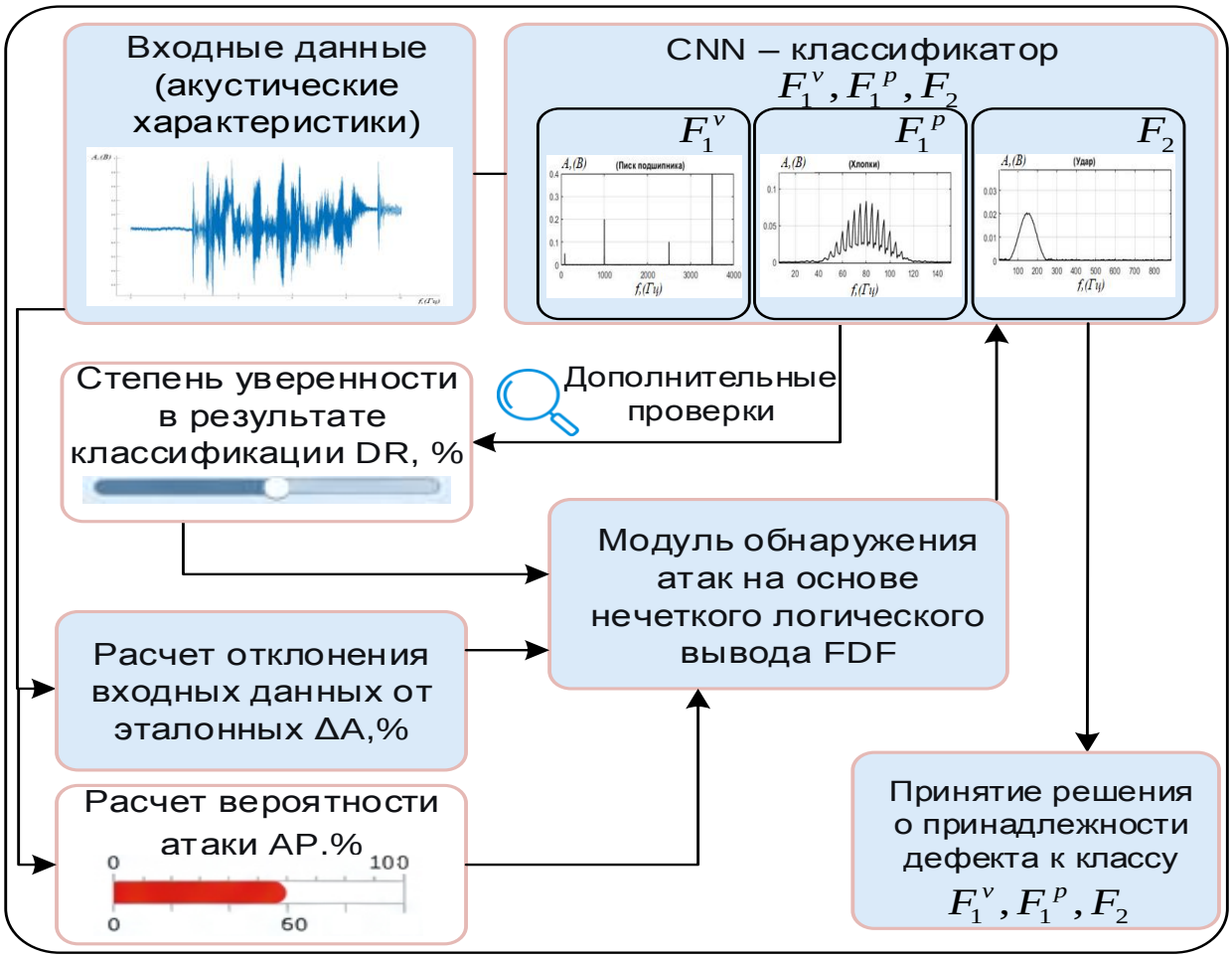


Рисунок 2 – Схема модуля классификации дефектов алгоритмом CNN

Модуль для определения аномалий использует параметры: DR, % - степень уверенности нейросети, ΔA, % - разница между эталонными данными и получаемыми данными предобработанных коэффициентов аудиособытий, AP, % - вероятность атаки, AD, % - обнаружение атаки (Attack Detected). Предварительный анализ осуществляется взаимодополняющими алгоритмами. Алгоритм на основе сверточной нейронной сети (CNN) выполняет обработку входных представлений и выявляет закономерности, недоступные традиционным методам диагностики. Нейросетевая архитектура позволяет эффективно работать с

многомерными данными и адаптироваться к новым паттернам неисправностей. Алгоритм на основе FDF реализует правила нечеткой логики, что обеспечивает устойчивость к неопределённости, размытости границ между классами и пересечению признаков различных типов отказов, а также устойчивость к атакам на данные нейросетевых моделей. Если коэффициент уверенности меньше порогового значения, то производится исследование программных дефектов на основе фаззинга. К фаззинг-тестированию была проведена предварительная проверка работы алгоритмов машинного обучения при выполнении функций ПАК.

Представлены сравнительные данные о степени уверенности (DR, %) алгоритмов - CNN и FDF при выявлении аппаратных и техногенных событий при корректности работы подсистем R_{f1}- R_{f4}. Подход позволяет корректно интерпретировать ситуации, когда признаки неисправности выражены неоднозначно. Структурная схема модуля анализа представлена на рисунке 3.

Таблица 1 –Входные параметры CNN (акустические признаки)

№	Расчетная формула	Параметры оборудования	Физический смысл
k_1	$k_1 = \alpha_1 f_{\text{ср}} + \beta_1$	$f_{\text{ср}}$-средняя частота спектра сигнала	Отражает общий наклон спектральной характеристики и смещение в сторону высоких или низких частот
k_2	$k_2 = \alpha_2 \Delta f + \beta_2$	Δf-разброс частот в спектре	Описывает "изогнутость" спектра (наличие пиков или впадин)
k_3	$k_3 = \alpha_3 (A_{\text{в}} - A_{\text{н}}) + \beta_3$	$A_{\text{в}}$-амплитуда на высоких частотах; $A_{\text{н}}$- амплитуда на низких частотах	Характеризует разницу амплитуд
k_4	$k_4 = \alpha_4 \Delta f_{\text{пик}} + \beta_4$	$\Delta f_{\text{пик}}$-ширина основного пика в спектре	Показывает ширину основного пика в спектре ($\Delta f_{\text{пик}}$).
k_5	$k_5 = \alpha_5 A_{\text{асим}} + \beta_5$	$A_{\text{асим}}$- асимметрия спектра	Отражает асимметрию спектра ($A_{\text{асим}}$ - разница амплитуд справа и слева от центральной частоты).
k_6	$k_6 = \alpha_6 A_{\text{шум}} + \beta_6$	$A_{\text{шум}}$- уровень шума	Связан с уровнем шума ($A_{\text{шум}}$ -средняя амплитуда на высоких частотах).
k_7	$k_7 = \alpha_7 N_{\text{гарм}} + \beta_7$	$N_{\text{гарм}}$- количество гармоник (пиков) в спектре	Показывает количество гармоник ($N_{\text{гарм}}$) в спектре.
k_8	$k_8 = \alpha_8 E_{\text{общ}} + \beta_8$	$E_{\text{общ}}$- общая энергия сигнала	Отражает общую энергию сигнала ($E_{\text{общ}}$- суммарная амплитуда спектра).
α_i – углы наклона спектральной характеристики; β_i- смещения спектральной характеристики (i=1..8)			

Модуль, представленный на рисунке 3, реализуется в виде ансамбля из трёх параллельных нейросетей, каждая из которых обучается на данных, соответствующих определённому типу неисправностей. В систему признаков включается набор акустических характеристик, обозначения которых приведены в таблице 1.

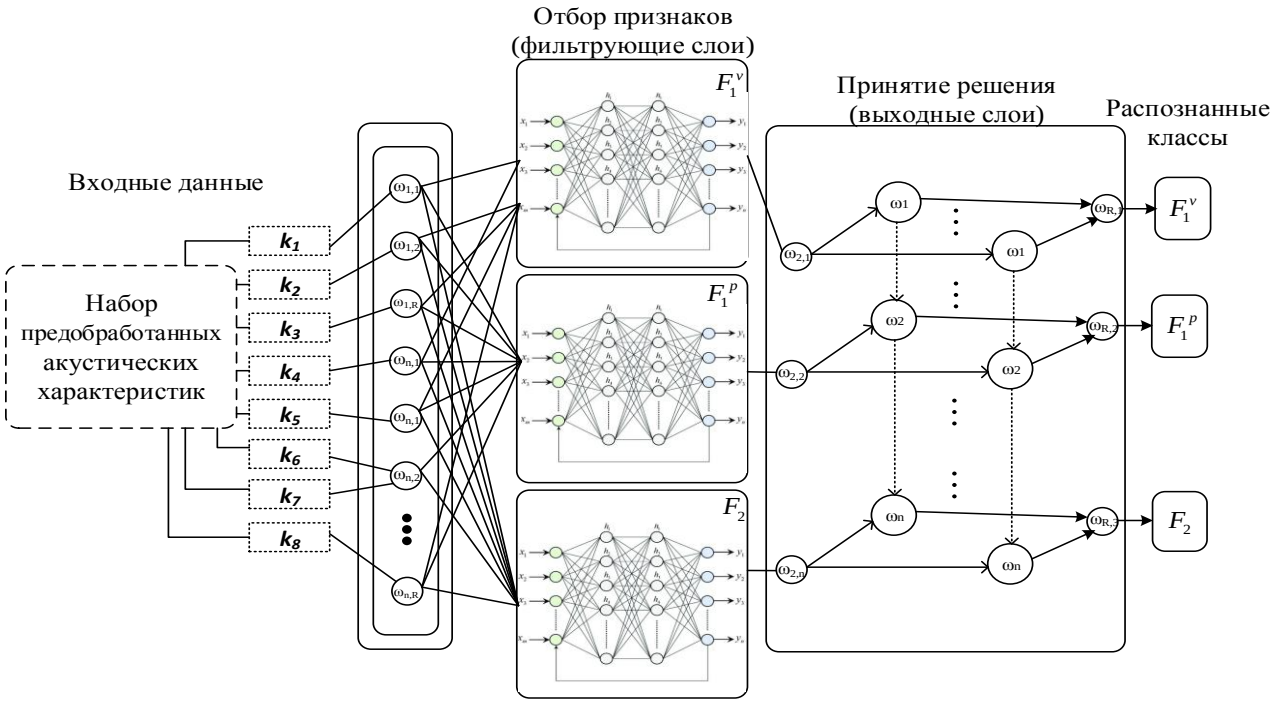


Рисунок 3 – Структурная схема модуля на основе анализа предобработанных акустических данных с помощью CNN

На этапе принятия решения каждая из трёх моделей формирует свою оценку вероятности наличия соответствующей неисправности. Система сравнивает полученные значения и выбирает класс с максимальной вероятностью. Дополнительно, по значению вероятностей можно оценивать уровень уверенности системы в полученном предсказании, что особенно важно в системах мониторинга, где допускается ручная верификация или запуск дополнительной проверки при низкой достоверности результата. Объединение результатов работы двух модулей FDF и CNN позволяет избежать ошибок, которые могут проявиться в результате умышленного воздействия искажений в данные на нейросеть (рисунок 4).

Для реализации алгоритма нечеткого логического вывода входные переменные показателей переводятся в значения нечетких лингвистических переменных блоками фаззификации *MF*1 - *MF*3. При выполнении фаззификации число, принадлежащего множеству действительных чисел, представляется в виде нечеткого числа. Полученные нечеткие переменные используются при выполнении операций, сформулированных в виде нечетких правил, реализуемых блоками *П*1-*П*9).

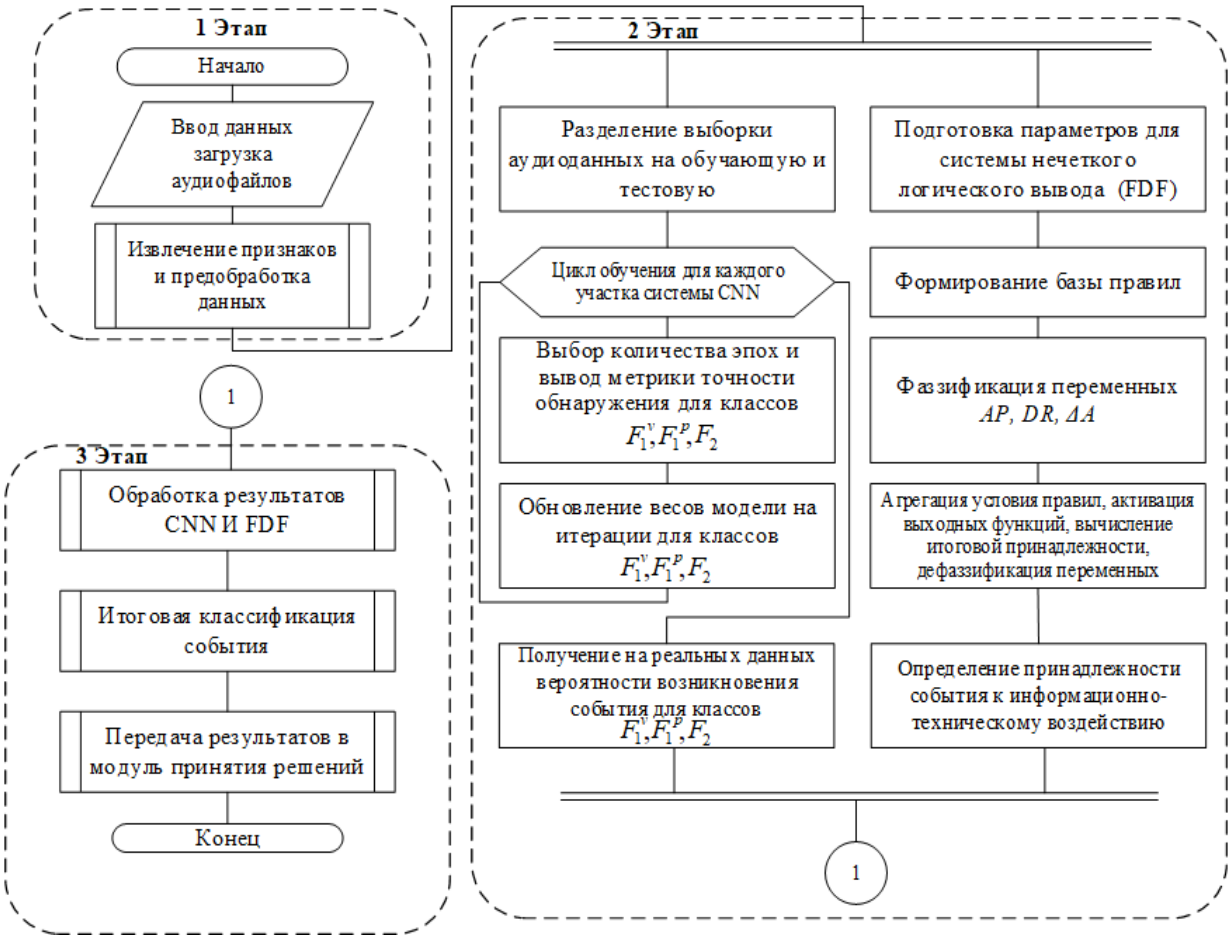


Рисунок 4 – Схема алгоритма анализа устройств подсистемы распределённого ПАК

Таблица 2 – Условия обнаружения атаки с помощью FDF

№	Условия обнаружения атаки
Наличие атаки (критические случаи)	
П1	***ЕСЛИ*** *(DR = L)* ***И*** *(ΔA = VL)* ***И*** *(AP = VL),* ***ТО*** *AD = P (AD* = 1)*
П2	***ЕСЛИ*** *(DR = L)* ***И*** *(ΔA = L)* ***И*** *(AP = VL),* ***ТО*** *AD = P (AD* = 1)*
П3	***ЕСЛИ*** *(DR = L)* ***И*** *(ΔA = VL)* ***И*** *(AP = L),* ***ТО*** *AD = P (AD* = 1)*
Высокая степень уверенности в наличии атаки (не требует проверки)	
П4	***ЕСЛИ*** *(DR = L)* ***И*** *(ΔA = M)* ***И*** *(AP = L),* ***ТО*** *AD = P (AD* = 1)*
П5	***ЕСЛИ*** *(DR = L)* ***И*** *(ΔA = L)* ***И*** *(AP = M),* ***ТО*** *AD = P (AD* = 1)*
П6	***ЕСЛИ*** *(DR = M)* ***И*** *(ΔA = L)* ***И*** *(AP = L),* ***ТО*** *AD = P (AD* = 1)*
Средняя степень уверенности в наличии атаки (требует проверки)	
П7	***ЕСЛИ*** *(DR = L)* ***И*** *(ΔA = S)* ***И*** *(AP = M),* ***ТО*** *AD = P (AD* = 1)*
П8	***ЕСЛИ*** *(DR = M)* ***И*** *(ΔA = M)* ***И*** *(AP = M),* ***ТО*** *AD = P (AD* = 1)*
П9	***ЕСЛИ*** *(DR = M)* ***И*** *(ΔA = S)* ***И*** *(AP = L),* ***ТО*** *AD = P (AD* = 1)*

Для фаззификации входных переменных диапазоны их изменения разбиваются на лингвистические термы (на основе экспертных оценок). При этом для каждого показателя используется 5 термов значений: очень малое *(VL),* малое *(L),* среднее *(M),* большое *(H) и* очень большое *(VH).* Для каждого из термов строится функция принадлежности $\mu(x)$ переменной x этому терму. Для задания центральных лингвистических термов входных переменных использована симметричная гауссова функция принадлежности (*gaussmf*), формируемая в соответствии с выражением $\mu(x) = e^{-(x-c)^2/(2\sigma^2)}$, где параметр с задает модальное

значение функции, а σ -ширину. Для задания крайних термов используются сигмоидальные функции принадлежности (*sigmf*), которые определяются в соответствии с выражением $\mu(x) = (1+e^{-a(x-c)})^{-1}$, где $a < c$. Параметр с определяет координату точки перегиба функции, а коэффициент a - наклон функции в этой точке. Заключение каждого правила задается в виде значения критерия обнаружения атаки *AD*, которое реализовано как одноэлементное нечеткое множество с точечной функцией принадлежности (таблица 2). Диапазон изменения переменной *R* также разбивается на 5 термов: нейтральное (*Z*), средне отрицательное (*NM*), средне положительное (*PM*), отрицательное (*N*) и положительное (*P*), (в соответствии с принадлежностью значения критерия этим термам принимается решение об окончании испытаний). Далее производится определение четкого числа критерия обнаружения атаки *AD* путем выполнения процедуры дефаззификации (обратного преобразования нечетких переменных в четкие). Вычисление критерия обнаружения атаки *AD* осуществляется путем определения взвешенного среднего в соответствии с выражением: $AD = \sum_{i=1}^{m} \mu(AD) AD_i (\sum_{i=1}^{m} \mu(AD))^{-1}$, где AD_i - значение выходной переменной для i-го терма с единичным значением степени принадлежности; $\mu(AD)$ -степень принадлежности к этому терму; m - число термов. Далее представлены результаты количественной оценки способности системы детектировать классы акустических событий. В условиях нормального режима работы система, использующая CNN, демонстрирует более высокую точность классификации техногенных событий, что подтверждается большим количеством корректно распознанных классов сигналов в каждой из подсистем. Количество ошибок 1 рода (1,5-2,8 %) и 2 рода (3,1-1,7 %) незначительно, что свидетельствует о пригодности разработанных алгоритмов для разграничения аппаратных и техногенных сбоев (таблица 3).

При наличии атак очевидно преимущество FDF, количество ошибок сократилось. Общепризнанными оценками качества работы классификатора являются метрики **точности** $J_p = G_p^+ / (G_p^+ + G_p^-)$ и **полноты** $J_r = G_p^+ / (G_p^+ + G_n^-)$ классификации. Отмечается, что на практике одновременное достижение максимальных значений Jp и Jr не всегда возможно, поэтому на основе полученных значений точности и полноты вычисляется значение **F-меры** – гармонического среднего между точностью и полнотой $F = 2JpJr/(Jp+Jr)$.

Таблица 3 – Результаты классификации дефектов сверточной нейросетью

N^a	F_k	*Показатели классификации*			*Ошибка первого рода, ε^I, %*	*Ошибка второго рода, ε^{II}, %*	N^{ce} *(при наличии атак)*	N^{ce} *(без атак)*
		Точность, J_p,%	*Полнота, J_r,%*	*F-мера,%*				
Классификация при высокой степени уверенности в отсутствии ИТВ								
10^5	F_1^v	97	96	97	2,8	3,1	295	345
	F_2	98	98	98	1,7	2,0	198	274
Классификация при низкой степени уверенности в отсутствии ИТВ и выявлении атак с применением FDF								
10^5	F_1^v	97	97	97	2,3	2,6	215	402
	F_2	98	98	98	1,5	1,7	148	375
N^a – общее количество экспериментов, F_k, – класс дефекта (F1- аппаратный отказ; F2- техногенное событие), N^{ce}-количество ошибок классификации, Δ^{ce}- разница в количестве выявленных дефектов, ИТВ - информационно-технические воздействия								

Однако при тестировании на моделированных неисправностях наблюдается обратная тенденция: алгоритм FDF превосходит CNN по количеству обнаруженных дефектов, что связано с его большей устойчивостью к искажениям и вариативности сигналов в условиях аномалий (таблица 4). Следует отметить, что эффективность нейросетевого алгоритма снижается при наличии атак на данные, что проявляется в ухудшении распознавания классов сигналов. После принятия решения о программном сканировании на основе полученного признака *DR* выполняется фаззинг. При таком подходе фаззинг-тестирование возможно осуществлять направленно. Для анализа использовались следующие метрики: покрытие кода (C, %) - количество логических путей и модулей, охваченных тестами, среднее время поиска неисправности как время от возникновения сбоя до его идентификации. Для фаззинга ставится задача увеличения объема обнаруженных ошибок за установленный период тестирования. $\sum_x b_x \to \max$, где b_x кумулятивный набор дефектов за счет применения тестовых наборов в распределенной программно-аппаратной системе для набора атакующих запросов на основе уязвимостей и дефектов устройств.

Таблица 4 – Выявлено дефектов при FDF и CNN

F_k	N^{ce} (при наличии атак)		Δ^{ce}, %	N^{ce} (без атак)		Δ^{ce}, %
	CNN	*FDF*		*CNN*	*FDF*	
F_1^v	295	215	31,3	345	402	15,2
F_2	198	148	28,9	274	375	31,1

Для $c_{i,j}$ - *i*-й сбой или отказ при выполнении *j*-го искажения данных; $A_{m(t)} = \{a_{q1}, a_{q2}, ..., a_{|a_{q|Q}|}\}$ - набора атакующих запросов на основе уязвимостей и дефектов устройств; $R_{m(t)} = \{r_{q1}, r_{q2}, ..., r_{|a_{q|Q}|}\}$ - наборов реализованных дефектов с помощью объединения оригинальных шаблонов (*O*) и модифицированных; $P = \{o_1, o_2, ..., o_{|O|}\} \cup \{g_1, ..., g_n\}$ - множество тестов программы для устройства ПАК.

Алгоритмы фаззинга с использованием моделей уязвимостей для проверки устройств ПАК на наличие актуальных программных ошибок опирается на данные из баз уязвимостей MITRE и OWASP, что обеспечивает его актуализацию в процессе тестирования. Алгоритм включает подготовку данных с определением режимов испытания устройства согласно шаблону спецификации *по шаблону 1*, алгоритм внесения отказов и сбоев в конечное устройство с изученной программной документацией по *шаблону 2*, с помощью техники фаззинга и алгоритмом определения тестовых испытаний по имитации неисправностей на основе нечеткого логического вывода. Тестирование устройства производится путем искусственного внесения ошибок в программное и аппаратное обеспечение проверяемых устройств на основе документации и статистики ошибок, которые были обнаружены в период ручного тестирования или разработки (рисунок 5).

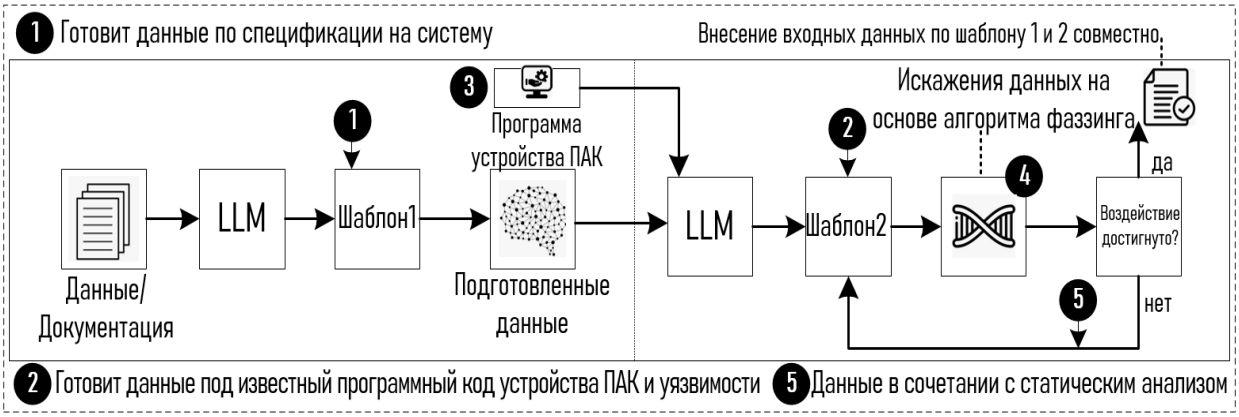


Рисунок 5 – Структурная схема конвейера генерации воздействий с использованием *LLM*

Конвейер тестирования стартует, на основе полученного контекста, *LLM* генерирует тестовые воздействия. Эти воздействия затем автоматически применяются к тестируемой системе. Реакция системы фиксируется и анализируется с использованием *CNN*, временного анализа и оценки отклонений в поведении и обнаружения потенциально успешных атак. Далее результаты наблюдения используются для генерации новых запросов и адаптации последующих тестовых воздействий, формируя замкнутый цикл: «генерация – исполнение – наблюдение – адаптация». Этот цикл обеспечивает эволюционное улучшение качества воздействия и повышение вероятности выявления уязвимостей. Работа алгоритма LLM для

повышения объема тестирования программы с использованием фаззинга строится следующим образом. Проводится сбор и предварительная обработка данных: собираются образцы входных сообщений, отчеты о работе системы, спецификации протоколов и протоколы взаимодействия. Эти данные используются для настройки языковой модели, чтобы адаптировать её под конкретную предметную область. Модель интегрируется в фаззинг-систему, которая вместо случайной генерации использует модель для создания более реалистичных и содержательных тестовых данных. Ключевое нововведение алгоритма LLM (L) определяет условную вероятность $P_L(i \mid C)$ для генерации нового индекса входа i при наличии контекста C (информация, доступная модели для принятия решения). Новый индекс входа (i_{new}) - данные, которые будут поданы в программу устройства ПАК. Контекст может включать исходный код программы устройства ПАК (полностью или частично), начальный набор валидных входов, историю предыдущих сгенерированных индексов входов $\{i_1, i_2, ..., i_{t-1}\}$, информацию обратной связи (покрытие кода, отчеты о сбоях). Так, на каждом шаге t фаззинга L генерирует новый номер входа i_t и вычисляется *вероятность* $P_L(i|C_t)$ для увеличения успешных запросов с имитацией ошибки: $\max_{\theta} E\left[\sum_{t=1}^{T} o(i_t)\right]$, *где* θ - параметры LLM (параметры запроса), для которого проводится оптимизация, i_t -вход, сгенерированный на шаге t согласно $P_L(i \mid C_t; \theta)$, $O(i_t)$ - результат проверки входа на эксплуатации ошибки, E - математическое ожидание, которое учитывает вероятностный характер генерации входов. Модель обучается на примерах корректных сообщений, передаваемых между компонентами системы, после чего генерирует тестовые данные, которые не только соответствуют синтаксису протокола, но и учитывают его использование.

В рамках разработанной системы, ориентированной на повышение эффективности выявления сложных и целенаправленных атак, применяется подход генерация с усилением (за счёт программной документации). Данный подход объединяет возможности семантического извлечения актуальной информации из специализированных структурированных баз знаний: базы данных уязвимостей (Common Vulnerabilities and Exposures - CVE), описания данных и этапов атак, т.е. последовательность команд, которые используют уязвимости в ПО и шаблоны аномального поведения с генеративными возможностями LLM. Это позволяет формировать высокоадаптивные и целенаправленные тестовые воздействия, предназначенные для анализа системы в условиях, приближённых к реальному функционированию (рисунок 6).

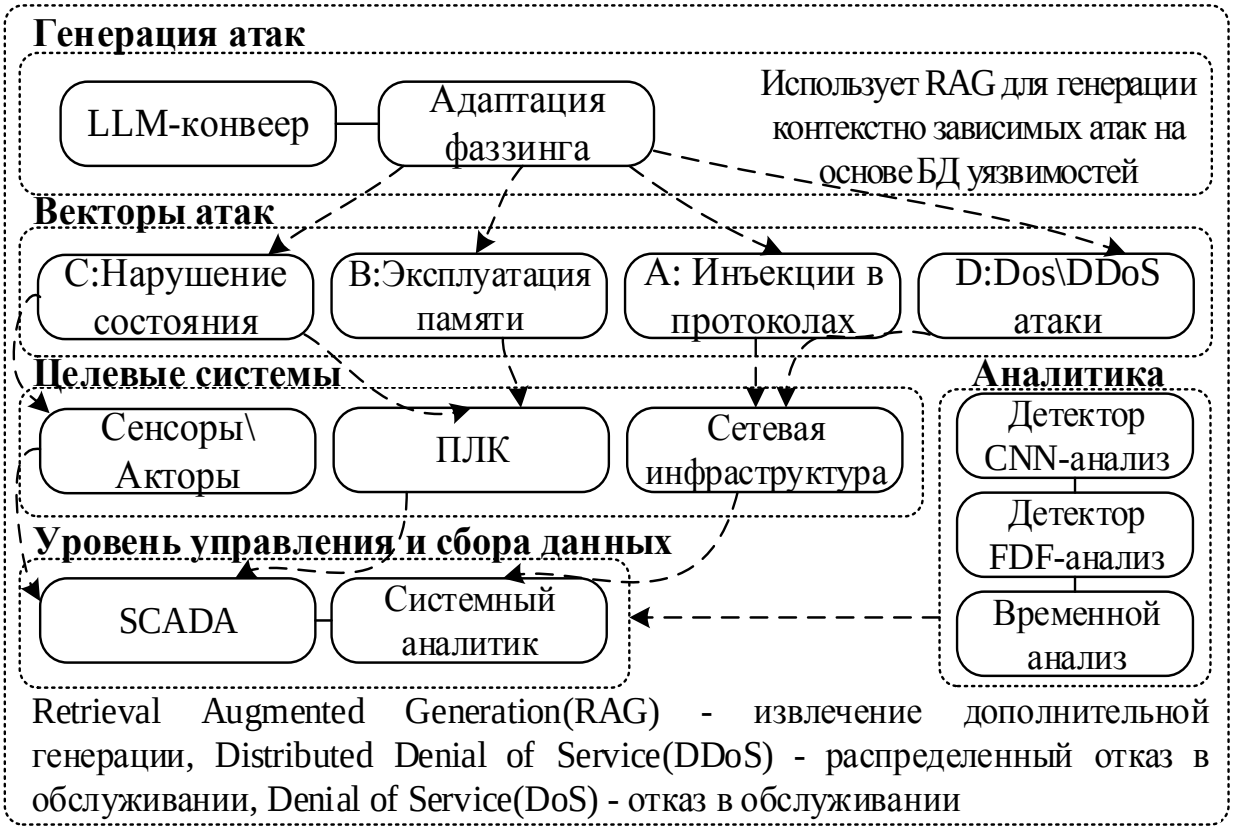


Рисунок 6 – Моделирование атак и обработки инцидентов в ПАК

При моделировании угроз в распределённом ПАК формируются четыре вектора атак, направленных на проверку подсистем распределенного ПАК. Каждый вектор моделируется с использованием генеративных возможностей алгоритмов LLM, в совокупности с фаззингом, ориентированным на индустриальные протоколы, данные управления и интерфейсы низкоуровневого взаимодействия. Вектор A охватывает инъекции кода и команд, включая SQL-инъекции, команды операционной системы и конструкции для проверки уязвимостей RCE и обхода процедуры проверки пользователя с целью тестирования реакции подсистем управления R_{f1} и связи R_{f2}. Вектор B используется для атак на память (переполнением буфера, подменой адресов возврата и внедрением кода) для проверки защиты подсистемы управления от манипуляций с критическими функциями. Вектор C моделирует нарушения допустимого состояния системы, включая изменение конфигураций, подмену телеметрии и управляющих сигналов, проверяя защиту и восстановление R_{f1}, R_{f4} и R_{f3}. Вектор D выполняет распределённые атаки DoS/DDoS, создающие перегрузку каналов связи и мониторинга, а также каскадные сбои, проверяя устойчивость к отказам и сбоям R_{f2} и R_{f3}. Совместная реализация этих векторов позволяет увеличить количество выявленных ошибок и ключевых уязвимостей распределенного ПАК .

В четвертой главе приведены результаты испытаний подсистем ПАК. Эксперимент проводился для распределённого ПАК с известным набором дефектов и направлен на исключение аппаратных и техногенных причин сбоев. Проводится

детальная проверка серверов, сетевых компонентов и систем хранения данных на предмет возможных неисправностей. Одновременно анализируются внешние воздействия, настройки среды и факторы, связанные с человеческим взаимодействием, с целью исключения внешних и аппаратных причин. Затем анализируются программные дефекты (таблица 5).

Таблица 5 – Общее количество программных ошибок для подсистем распределенного ПАК

Подсистема	Ошибки ПО (без внешних воздействий),%	Ошибки ПО при техногенных воздействиях,%	Ошибки ПО при ИТВ,%	Время доведения до оператора, ч
Стандартный алгоритм фаззинга /Алгоритм фаззинга с LLM коррекцией				
R_{f1}	42 / 67	10 / 30	7 / 10	8 / 4
R_{f2}	51 / 67	15 / 20	2 / 5	12 / 7
R_{f3}	57 / 77	5 / 20	8 / 15	6 / 3
R_{f4}	45 / 74	4 / 8	2 / 5	12 / 10
R_{f1}- управления, R_{f2}- связи, R_{f3}- мониторинга, R_{f4}-сенсорный контур				

Средние значения прироста покрытия для подсистем: R_{f1} – 25%, R_{f2} – 16%, R_{f3} – 20%, R_{f4} – 29%. Три подсистемы демонстрируют устойчивый рост покрытия с увеличением времени фаззинга, в то время как R_{f4} показывает меньший рост. Для оценки были выбраны четыре подсистемы ПАК (рисунок 7).

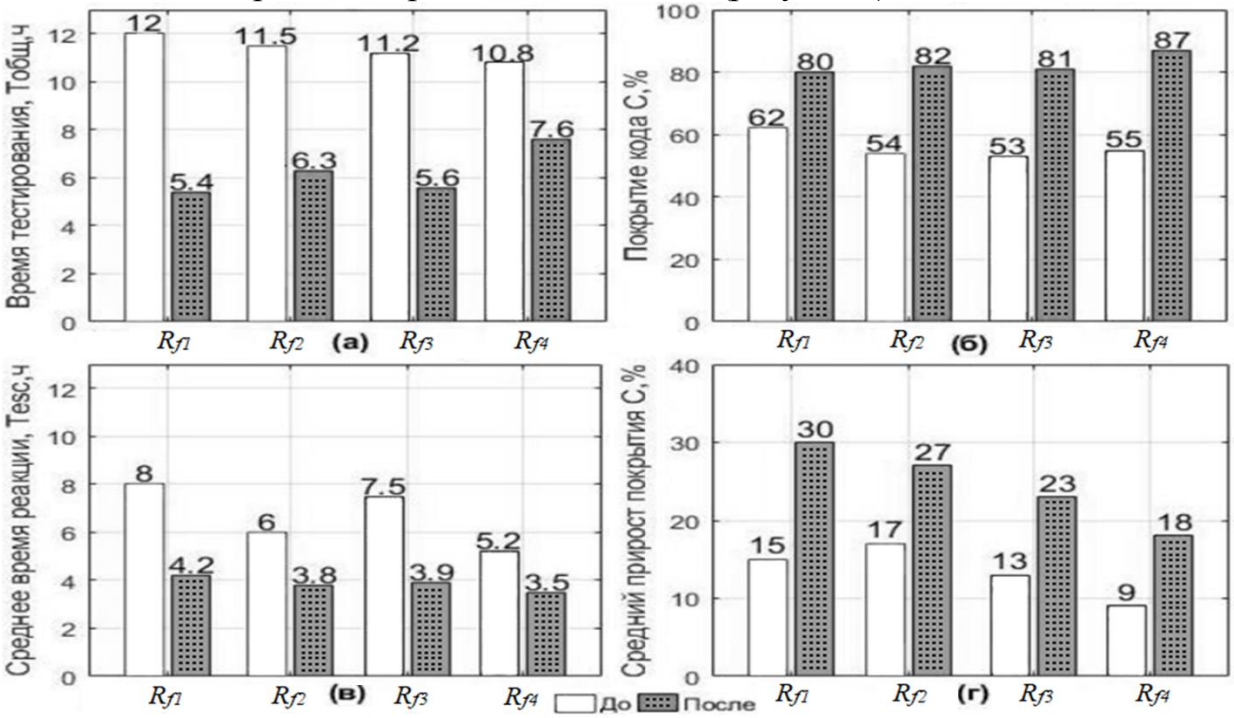


Рисунок 7 – Сравнение результатов тестирования подсистем ПАК до и после внедрения фаззинга: а) – время тестирования, б) – покрытие кода , в) – время реакции оператора, г) – прирост покрытия кода

Представленные обобщённые результаты, полученные в процессе имитации отказов и последующего тестирования. На диаграмме видно, что время тестирования до и после оптимизации для подсистем R_{f1}-R_{f4}. По результатам удалось достичь сокращения времени реакции оператора, увеличить покрытие кода и найти все заданные уязвимости.

ЗАКЛЮЧЕНИЕ

В результате проведённых исследований получены новые научные и практические результаты, направленные на повышение устойчивости к отказам и сбоям распределённых программно-аппаратных комплексов, состоящих из подсистем разнородной структуры.

1. В результате анализа работ, посвященных вопросам проявления новых свойств распределенных программно-аппаратных комплексов при реконфигурации, внешних воздействиях, отказах и сбоях, выявлено, что известные методики обнаружения неисправностей в распределенных ПАК недостаточно эффективны и требуют усовершенствования. Установлено, что для проектируемых ПАК основной проблемой обеспечения работоспособности является выявления отказов и сбоев, связанных с дефектами аппаратных и программных средств. Выявлена необходимость разработки моделей и алгоритмов обнаружения и классификации дефектов аппаратных и программных средств ПАК с применением современных интеллектуальных технологий.

2. Разработаны модель и алгоритм для определения видов дефектов с использованием сверточной нейронной сети, обрабатывающей временные и спектральные характеристики акустических сигналов с датчиков ПАК в условиях информационно-технических воздействий. Установлено, что классификация при высокой степени уверенности в отсутствии ИТВ имеет высокую точность: $J_{p,}$ = 97% (при обнаружении аппаратных дефектов) и $J_{p,}$ = 98% (при обнаружении техногенных событий).При низкой степени уверенности в отсутствии ИТВ и дополнительном привлечении нечеткого логического вывода FDF для выявления ИТВ (кибератак) сверточная нейронная сеть позволила увеличить количество выявленных дефектов. Так количество найденных аппаратных дефектов увеличилось (с 345 до 402 без действия атак на ПАК), также как и количество установленных техногенных событий (общее количество испытаний N^a =10000).

3. Разработан алгоритм имитации неисправностей с применением техники фаззинга для обнаружения дефектов программных средств распределенного программно-аппаратного комплекса, позволяющий увеличить объем тестирования (по сравнению с традиционными методами). Так, при количественной оценке динамики покрытия кода программ, осуществляющих обработку и обмен данными в ПАК, получено, что среднее значения прироста покрытия кода (разница между 12-часовым и исходным уровнем) составляет 25% (с 19.1% до 29.4%).

Так же сократились временные затраты на тестирование: понадобилось от 4 до 12 часов для проверки четырех подсистем ПАК (вместо 8-24 часов, необходимых для тестирования традиционными методами).

4. Разработаны модель и алгоритм выявления критических уязвимостей в ПО распределенного ПАК с помощью нейросетевой большой языковой модели LLM для совместного использования с техникой направленного фаззинга на потенциальные уязвимости при тестировании ПАК, позволяющего повысить объем выявляемых ошибок в ПО. За счет обучения на технической документации достигнуто расширение покрытия кода на 25% (по сравнению с традиционным фаззингом). В результате применения разработанного алгоритма обеспечено ускоренное выявление программных дефектов, вызванных ими отказов и сбоев, что позволяет снизить время доведения до оператора системы на 57 % и своевременно передавать информацию о неисправностях.

5. Создан программный комплекс на основе предлагаемых моделей и алгоритмов для проведения экспериментальных исследований, с использованием которого подтверждено повышение объема выявляемых дефектов. Выполненная автоматизация проведения испытаний устройств связи позволила увеличить количество выявленных дефектов ПАК по сравнению с классическим тестированием на 47% (выявлен 61 дефект вместо 41). Внедрение разработанных программно-алгоритмических средств тестирования позволило обнаружить семь ранее не выявленных уязвимостей программного обеспечения (на 25% больше), влияющих на работу подсистем распределенного комплекса.

ОСНОВНЫЕ РАБОТЫ, ОПУБЛИКОВАННЫЕ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в научных журналах, рекомендованных ВАК при Минобрнауки России

1. **Панков, И.А.** Выявление дефектов при тестировании алгоритмов цифровых устройств на базе ПЛИС / И.А. Панков // Известия Тульского государственного университета. Технические науки – 2023. №11. – С.277-280.

2. Панков, Д.А. Обнаружение системных дефектов цифровых устройств при имитации неисправностей с применением фаззинга / Д.А. Панков, **И.А. Панков** // Надежность. 2023. – Т.23 №4. – С. 51-58.

3. Панков, Д.А. Автоматизация разработки и тестирования цифровых систем связи с многоуровневой архитектурой / Д.А. Панков, **И.А. Панков**, Л.А. Денисова // Автоматизация в промышленности. – 2023. № 1. – С.31-35.

4. **Панков, И.А.** Количественные и качественные методы оценки надежности автоматизированных систем управления / **И.А. Панков**, А.П. Панков, Д.А. Панков // Известия Тульского государственного университета. Технические науки. – 2024. – № 5. – С.198-203.

5. Панков, А.П. Перспективы использования FMEA-анализов для высокоответственных технических систем / А.П. Панков, Д.А Панков, Ю.П. Похабов, **И.А. Панков** // Известия Тульского государственного университета. Технические науки – 2024. –№ 3. – С.26-31.

6. **Панков, И.А.** Выявление системных неисправностей в программно-аппаратных комплексах на основе интеллектуальных технологий / И.А. Панков, А.П. Аверченко, Д.А. Панков // Надежность. – 2025. – Т.25 №4. – С. 61-68.

Государственная регистрация программ для ЭВМ

7. Свидетельство о государственной регистрации программы для ЭВМ № 2023612691 Российская Федерация. Модуль сопряжения; заяв. 25.01.2023; опубл. 07.02.2023 **/ И.А. Панков**; заявитель и патентообладатель АО ОНИИП.

8. Свидетельство о государственной регистрации программы для ЭВМ № 2025619291 Российская Федерация. УК в РПДУ; заяв. 03.04.2025; опубл. 15.04.2025 / **И.А. Панков**; заявитель и патентообладатель АО ОНИИП.

Статьи в материалах конференций

9. **Панков, И. А**. Автоматизация проектирования алгоритмов ЦОС без ухудшения качества выходного сигнала / И. А. Панков, Е. Р. Мирхайдаров // Нанотехнологии. Информация. Радиотехника (НИР-22) : материалы Регион. молодеж. науч.-практ. конф. (Омск, 21 апр. 2022 г.) / Ом. гос. техн. ун-т. – Омск : Изд-во ОмГТУ, 2022. – С. 37–40.

10. **Панков, И. А.** Ускорение поиска дефектов цифровых устройств / И. А. Панков // Мехатроника, автоматика и робототехника. – 2023. – № 11. – С. 180–183.

11. **Панков, И. А.** Выявление дефектов цифровых устройств на базе ПЛИС / И. А. Панков // Информационные технологии и автоматизация управления: материалы XIV Всерос. науч.-практ. конф. студентов, аспирантов, работников образования и промышленности (Омск, 26–27 мая 2023 г.) / Ом. гос. техн. ун-т. – Омск : Изд-во ОмГТУ, 2023. – С. 171–173.