

REDS:

Телекоммуникационные устройства и системы

№1

2026

СОДЕРЖАНИЕ

Александров А.Е., Рузин Н.И. ФУНКЦИОНАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ДИАГНОСТИКИ ЭЛЕКТРОТЕХНИЧЕСКОГО СОСТОЯНИЯ БПЛА С ПОМОЩЬЮ КОНТРОЛЬНО ПРОВЕРОЧНОЙ АППАРАТУРЫ	4
Бессолицын А.Д., Гадасин Д.В., Галицкий М.В., Гадасин Д.Д. АЛГОРИТМ АДАПТИВНОЙ БАЛАНСРОВКИ ДЛЯ РАСПРЕДЕЛЁННЫХ СИСТЕМ С СИЛЬНО СКОШЕННОЙ НАГРУЗКОЙ	10
Караваева Э.А., Ковтун И.И. ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ ИНСТРУМЕНТОВ В ПРОЦЕССЕ РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ПОТОКАМИ И ПРОИЗВОДСТВЕННЫМИ ПРОЦЕССАМИ ТКАЦКОЙ ФАБРИКИ	17
Нетес В.А. РЕКОМЕНДАЦИИ МСЭ-Т ПО ПРИМЕНЕНИЮ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ЭКСПЛУАТАЦИИ И УПРАВЛЕНИЯ ЭЛЕКТРОСВЯЗЬЮ	24
Андрюшина М.Н., Рожков С.А. РАСПОЗНАВАНИЕ ПЭТ-ОБЪЕКТОВ В ПОТОКЕ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ	30
Кархили Ханин, Давуд Таммам, Степанов С.Н. ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ РАДИОРЕСУРСОВ И ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ ДЛЯ РАЗНОРОДНОГО ОБРАЗОВАТЕЛЬНОГО ТРАФИКА В ПЕРСПЕКТИВНЫХ БЕСПРОВОДНЫХ СЕТЯХ	38
Фатхулин Т.Д., Мяличева А.А., Сахарова А.М. ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОГНОЗИРОВАНИЯ ДЕФЕКТОВ В ИСХОДНОМ КОДЕ С ПОМОЩЬЮ МОДИФИЦИРОВАННОГО АЛГОРИТМА XGBOOST	44

ФУНКЦИОНАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ДИАГНОСТИКИ ЭЛЕКТРОТЕХНИЧЕСКОГО СОСТОЯНИЯ БПЛА С ПОМОЩЬЮ КОНТРОЛЬНО-ПРОВЕРОЧНОЙ АППАРАТУРЫ

Александров Александр Евгеньевич

*Московский технический университет связи и информатики,
профессор кафедры МКиИТ, к.т.н., профессор, Москва, Россия,
femsystem@yandex.ru*

Рузин Николай Игоревич

*Московский технический университет связи и информатики», магистрант, Москва, Россия
flegoin@yandex.ru*

Аннотация

В данной работе рассматривается архитектура функционального программного обеспечения (ФПО), решающего задачу диагностики электротехнического состояния различных серийно изготавливаемых изделий класса беспилотных летательных аппаратов (БПЛА) с применением контрольно-проверочной аппаратуры (КПА). Особое внимание уделено обеспечению надежного и безопасного функционирования ФПО на отечественных операционных системах специального назначения (ОС СН).

Ключевые слова:

контрольно-проверочная аппаратура, технический контроль, диагностика изделий БПЛА, неразрушающий контроль, автоматизация проверки электротехнического состояния.

Введение

В последние годы наблюдается активный рост применения беспилотных летательных аппаратов (БПЛА) в различных областях – в мониторинге транспортного движения [3], контроле состояния сельхозугодий [4], инфраструктуры [5], геологической разведке, археологии, метеорологии в решении военных задач [6]. За счет широкой области применения, существует большое видовое разнообразие БПЛА.

Производство различных видов гражданских и военных БПЛА в России активно растет и, по прогнозам экспертов, сохранит темпы прироста более чем на 20% в год на ближайшие 5 лет [7]. Изготовленные БПЛА хранятся в ангарах, где на них воздействуют разрушающие внешние факторы: пыль, избыточная влажность, плесень, грибок, грызуны, насекомые. Также, со временем происходит деградация используемых аккумуляторов и конденсаторов, рассыхание резиновых элементов, коррозия контактов.

Таким образом, стоит задача оптимизации и автоматизации технического обслуживания серийно изготавливаемых различных видов изделий, в частности проведения первичного и периодического (регламентного) контроля БПЛА и входящих в него элементов. Технический контроль – комплекс мероприятий, направленных на проверку исправности, работоспособности и готовности БПЛА к выполнению полетных заданий. Технический контроль является обязательным этапом технического обслуживания.

Одним из этапов технического контроля БПЛА является диагностика его электротехнического состояния – проверки работоспособности электронных систем, цепей питания, датчиков и управляющей аппаратуры. В частности, диагностика подтверждает, что проверяемое изделие функционирует ожидаемым образом в соответствии с конструкторской и эксплуатационной документацией. Она может производиться как для отдельного прибора из состава БПЛА, так и для всего изделия в совокупности. Решение о пригодности БПЛА для выполнения целевых задач принимают на основе результатов серии проведенных диагностических процедур.

Диагностика электротехнического состояния многоэлементного БПЛА с помощью средств измерений (вольтметров, амперметров, осциллографов) вручную не оптимальна. Чтобы автоматизировать данный процесс, применяют специализированную контрольно-проверочную аппаратуру (КПА) [2] в связке со функциональным программным обеспечением (ФПО), реализующим логику поэлементной проверки изделия. Данное ФПО является предметом исследования.

ФПО должно в определенные моменты времени и с определенной последовательностью подавать и снимать питание, отправлять команды по различным каналам информационного обмена и снимать показания измерительных приборов с КПА. Набор подобных действий является проверкой изделия или одной из его составных частей.

Результаты исследований

Целевая платформа

ФПО КПА предназначено для функционирования на вычислительных средствах КПА. КПА может быть физически подключено к проверяемому изделию через технологический канал и предоставляет следующие технические возможности:

- смыкание и размыкание силовых и сигнальных цепей;
- подача и снятие питания с цепей;
- измерение сопротивления и напряжения между контрольными точками технологического канала;
- опровка релейных (аналоговых) команд;
- прием релейных (аналоговых) сигналов;
- измерение температуры в приборном отсеке;
- прием и отправка сообщений на некоторые приборы из состава БПЛА по каналам Ethernet и МКИО [1] (мультиплексному каналу информационного обмена).

ФПО КПА использует данные возможности для реализации логики проверок. Однако, в проверяемых БПЛА не все приборы доступны для КПА напрямую, некоторая часть приборов может быть связана только посредством транзита через другие приборы. Это связано с тем, что разработчик БПЛА стремится упростить внутреннюю кабельную сеть, оптимизировать расходы материалов и уменьшить итоговую массу БПЛА.

Таким образом, ФПО КПА, посредством обмена аналогово-цифровыми сигналами и управлением наземными источниками питания, должно:

- идентифицировать факт подключения посредством технологического канала БПЛА к КПА;
- получать сведения о подключенном БПЛА, например, заводской номер или модификацию;
- в зависимости от модификации отображать для оператора список доступных проверок;
- проверять наличие недеklarированных соединений в цепи перед подачей питания (с помощью измерения сопротивления);
- фиксировать отклонения в работе проверяемого изделия или его составной части;
- устанавливать обмен с приборами, в том числе через транзитное соединение;
- сохранять и отображать историю проверок.

Архитектура ФПО КПА

Для описания архитектуры ФПО КПА, необходимо выявить сущности, с которыми оно взаимодействует. На рис. 1 представлена структурная схема КПА с подключенным изделием.

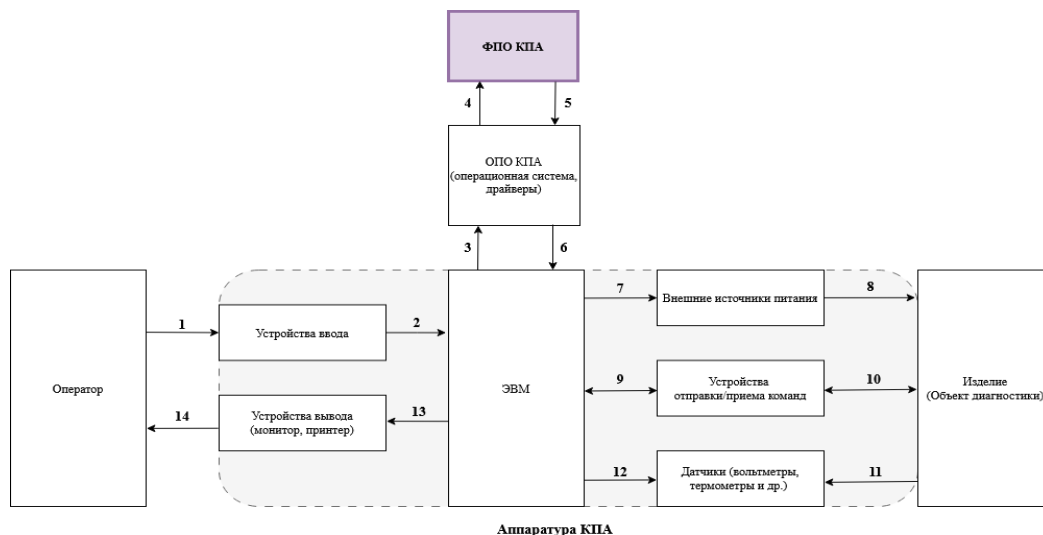


Рис. 1. Структурная схема КПА

Взаимодействие ФПО КПА с аппаратурой происходит через общее программное обеспечение (ОПО), включающее в себя операционную систему (ОС) и комплект драйверов, достаточных для управления всей периферией аппаратуры КПА. Связи 3, 4, 5, 6 являются программными и определяются установленным программным интерфейсом (API).

Оператор, через графический пользовательский интерфейс (ГПИ), предоставляемый ФПО КПА, определяет объем диагностических связей (связь 1). Полученный от оператора объем, через ОПО КПА, передается в ФПО (связи 2, 3, 4). При поступлении команды от оператора на начало диагностики, ФПО КПА последовательно и с установленным интервалом времени опрашивает команды на подачу питания изделия или его составных частей (связь 7), а также начинает обмен по аналоговым и цифровым каналам, предоставляемым устройствами опправки и приема команд (связи 9, 10). Периодически происходит съём ответных команд (связь 9) и данных с датчиков (связи 11, 12). Если все полученные от изделия команды и данные с датчиков корректны, то диагностируется исправность изделия. Ход диагностики и её результаты отображаются на ГПИ ФПО КПА в виде отчета, и предоставляются оператору (связи 13, 14). По запросу от оператора, отчет может быть распечатан на принтере.

Упрощенная структурная схема ФПО КПА приведена на рис. 2.

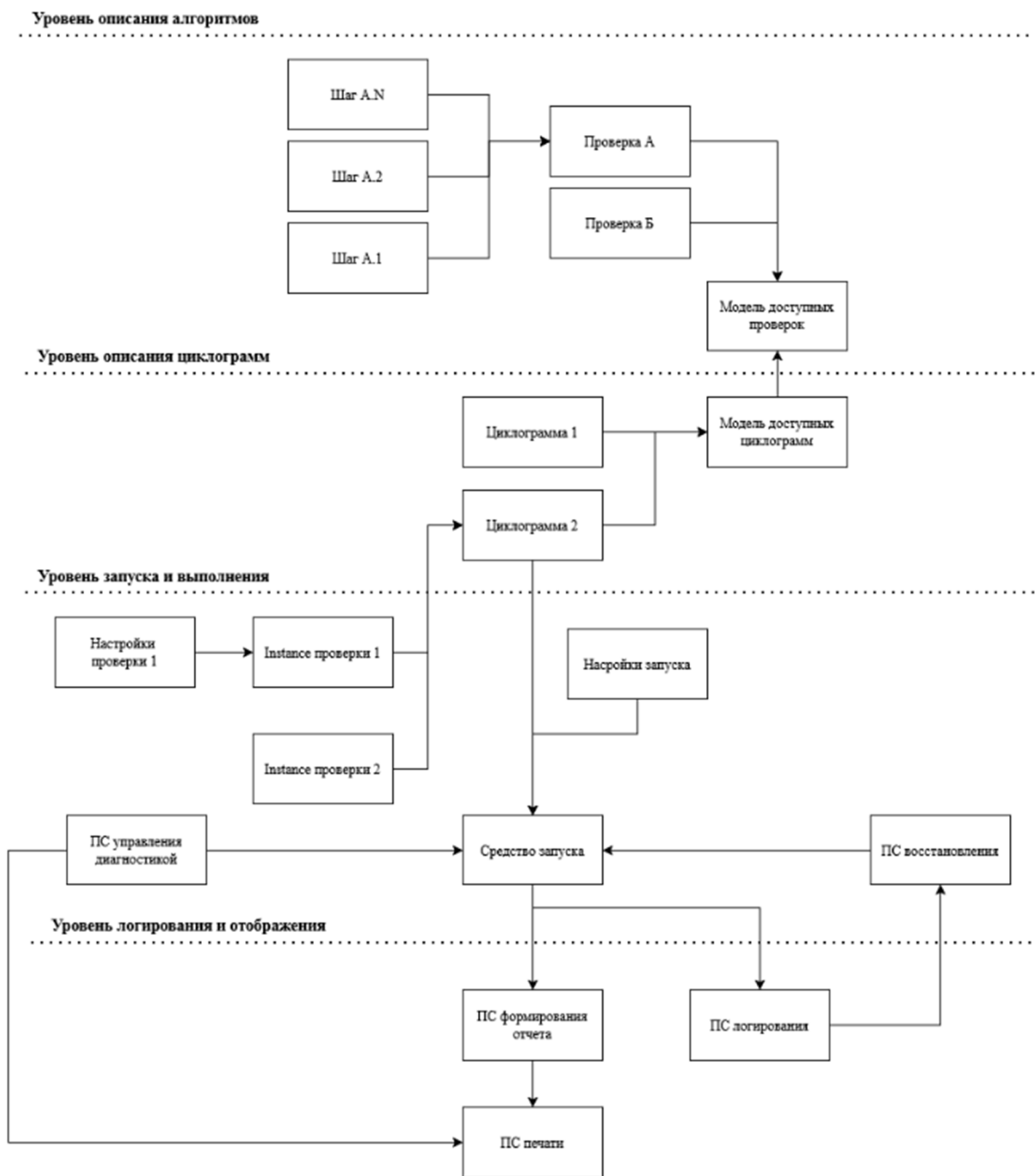


Рис. 2. Структурная схема ФПО КПА

Компоненты ФПО КПА можно разделить на 4 уровня по степени абстракции: уровень описания алгоритмов, уровень описания циклограмм, уровень запуска и выполнения диагностической проверки, а также уровень логирования и отображения.

Первый уровень хранит уникальные процедуры, реализующие алгоритмы. Каждый алгоритм представляет собой набор последовательно выполняемых шагов. Каждый шаг может завершиться без ошибок (норма) или с ошибкой (ненорма). Ошибка возникает, если в рамках выполнения шага, на отправленную команду диагностируемое изделие не сформировало корректный и ожидаемый ответ, или датчик зафиксировал превышение контролируемого параметра. В случае, если один из шагов вернул ненорму, процедура завершается по ненорме (в противном случае, проверка завершается по норме).

Каждая процедура представляет собой динамически подключаемую библиотеку с заданным программным интерфейсом. При запуске ФПО КПА последовательно происходит:

- 1) контроль факта подключения БПЛА к КПА;
- 2) идентификация модели подключенного БПЛА;
- 3) считывание всех доступных проверок из заданного места в файловой системе (ФС) и определение набора доступных проверок для считанной модификации изделия.

Процедуры могут иметь графические виджеты, посредством которых оператор может их настроить. Если процедура может быть настроена, то на интерфейсе ФПО КПА отображается кнопка, при нажатии на которую открывается типовое окно настроек проверки. Введенные настройки будут относиться не к самой процедуре, а к её образу (instance), поражаемому на уровне запуска и выполнения.

Процедуры разделены по назначению. На текущий момент выделены следующие виды процедур:

- процедуры непосредственной проверки;
- процедуры считывания ПО с прибора из состава изделия;
- процедуры записи ПО в прибор из состава изделия.

Все процедуры имеют общий интерфейс логирования хода выполнения, но результаты их выполнения отображаются по-разному. Так, например, после считывания ПО отображается окно, на котором выводится контрольная сумма (КС) и версия считанного ПО.

На следующем уровне несколько процедур объединяются в циклограмму. Циклограмма в данном контексте – набор процедур, которые выполняются последовательно. Циклограммы относятся к определенной модификации изделия и могут быть собраны из проверок, доступных для этой модификации. Если одна из проверок циклограммы возвращает ненорму, вся циклограмма считается выполненной с ошибкой. Каждая циклограмма обладает собственным именем и дополнительными параметрами, влияющими на то, как она будет запущена. Если оператор выберет циклограмму, в которой указан идентификатор несуществующей проверки, ФПО КПА выведет ошибку на экран. Циклограммы могут быть созданы, удалены и отредактированы посредством ГПИ ФПО КПА через автономное средство редактирования циклограмм. Это действие доступно в случае, если оператор обладает расширенными правами. При запуске ФПО КПА, описания циклограмм подгружаются и сохраняются в модели в доступном для верификации и запуска виде.

Оператор может запустить либо одну из имеющихся процедур напрямую, либо выбрать подготовленную циклограмму. В первом случае, формируется безымянная циклограмма, состоящая из одной процедуры.

На уровне запуска и выполнения, по каждой процедуре, входящей в выбранную циклограмму, формируется её образ. В отличие от процедуры первого уровня, образ хранит индивидуальные настройки и текущее состояние выполнения. Такой подход использует паттерны проектирования Fabric и Decorator [8], когда по одному объекту формируется другой объект и ему присваиваются дополнительные свойства. Выделение дополнительного слоя абстракции позволяет отделить алгоритм процедуры от его реализации, в том числе позволяет сделать так, чтобы у каждой проверки были собственные значения настроечных параметров.

После того, как выбранная циклограмма была верифицирована, а входящие в нее процедуры настроены, оператор может запустить выполнение. Процедуры последовательно выполняются в отдельном потоке с использованием методов асинхронного программирования. Выполнение может быть прекращено в одном из следующих случаев:

- оператор прервал выполнение через ГПИ ФПО КПА;
- зафиксировано превышение одного из измеряемых параметров (напряжение, ток, температура);
- алгоритм процедуры выдал сообщение о немедленном прекращении выполнения.

В таком случае, текущий шаг выполняется до конца, затем выполнение циклограммы завершается, и оператор получает сообщение с указанием причины остановки.

В ФПО КПА предусмотрен пошаговый (отладочный) режим. После того, как будет выполнен первый шаг первой процедуры в циклограмме, выполнение приостановится до того момента, пока оператор не подтвердит продолжение выполнения процедуры. Если необходимо, оператор перед паузой может перейти к любому из предыдущих шагов. Такой режим необходим в случае, если перед проверкой необходимо провести юстировку составных частей проверяемого изделия.

На следующем уровне ход выполнения и результаты выполнения отображаются на экран в виде отчета и графиков. Также, происходит детерминированная запись хода выполнения на ПЗУ ЭВМ из состава аппаратуры КПА. Запись происходит как в человеко-читаемом виде, так и в виде, который может быть загружен при восстановлении системы в случае аварийного завершения работы ФПО.

Графический интерфейс ФПО КПА

В соответствии с техническими требованиями, ФПО КПА должно функционировать на отечественных ОС, в частности на ОС СН Astra Linux [8]. Это значит, что при разработке перечень возможных для использования программных средств и библиотек ограничен. В этом случае был проведен анализ различного ПО как на уровне взаимодействия с ОС, так и для разработки интерфейса. Было принято решение использовать только те программные компоненты, которые входят в состав репозитория дистрибутива ОС Astra Linux, так как только в этом случае можно добиться совместимости и успешно пройти сертификацию.

Для разработки основной логики ФПО и реализации проверок был использован язык программирования C++, так как он имеет наибольшее распространение в команде разработки.

Авторами активно применялся фреймворк Qt5, который является базовым для основной в ОС Astra Linux графической оболочки fly (базируется на графической оболочке KDE).

Qt5 позволяет формировать графический интерфейс двумя способами – с помощью QtWidgets и с использованием QML.

При разработке ФПО КПА был использован хотя и устаревший, но более надежный подход с использованием QtWidgets. При определении способа вывода были выделены следующие преимущества:

- прямая интеграция с C++ кодом;
- зрелость экосистемы и документации;
- надежность для промышленных приложений.

Поставка

Аппаратура КПА является серийно-изготавливаемым изделием, поэтому процесс установки ФПО КПА должен быть автоматизирован. Дополнительным требованием является требование хранения даты, времени сборки ФПО КПА и контрольной суммы (КС) устанавливаемого образа.

Дата, время сборки и КС образа являются сведениями об ФПО КПА. Они хранятся в отдельном файле, который формируется всякий раз при установке ФПО КПА.

Так как дистрибутив ОС Astra Linux базируется на open-source дистрибутиве Debian, то наиболее логичным является использование пакетного менеджера dpkg. ФПО КПА поставляется в виде deb-пакета. Это упрощает и автоматизирует процесс установки, обновления и удаления.

Формат deb-пакета предусматривает наличие в своем составе скриптов на языке bash, которые запускаются в зависимости от события. В deb-пакете ФПО КПА используются скрипты, которые запускаются перед установкой, после установки и при удалении.

Заключение

В результате проведенного исследования была разработана архитектура функционального программного обеспечения для диагностики электротехнического состояния серийно изготавливаемых БПЛА, а также был реализован прототип для отработок на аппаратуре КПА.

Основные научные и практические результаты работы заключаются в следующем:

1. Разработана архитектура ФПО, которая интегрирует функции управления КПА, обработки сигналов и диагностики неисправностей.
2. Предложены решения для организации хранения, обработки и визуализации диагностических данных, обеспечивающие оперативный доступ к информации и поддержку принятия решений об исправности БПЛА;
3. Обеспечена надежность функционирования ФПО за счет его адаптации к работе под управлением отечественных операционных систем специального назначения (ОС СН), что соответствует требованиям к критически важным системам.

Внедрение разработанного ФПО позволяет автоматизировать процесс диагностики, снизить влияние человеческого фактора и повысить общую надежность эксплуатации БПЛА за счет своевременного выявления дефектов. Перспективы дальнейших исследований видятся в расширении доступных процедур проверок, внедрении графического языка формирования процедур.

Литература

1. ГОСТ Р 52070-2003. Интерфейс магистральный последовательный системы электронных модулей. Общие требования. – Введ. 2004-01-01. – М.: Стандартинформ, 2003
2. *Савин А., Губа В., Быкова О.* Контрольно-проверочная аппаратура для тестирования навигационных приёмников космических аппаратов // Современная электроника : электрон. журн. 2014. № 9. С. 46-47. URL: <https://www.cta.ru/articles/soel/2014/2014-9/4313/> (дата обращения: 19.01.2026)
3. *Дронова О. Б.* Современные возможности применения беспилотных летательных средств для обеспечения контроля безопасности дорожного движения // Научный вестник ОрЮИ МВД России им. В. В. Лукьянова. 2023. №1 (94). URL: <https://cyberleninka.ru/article/n/sovremennye-vozmozhnosti-primeneniya-bespilotnyh-letatelnyh-sredstv-dlya-obespecheniya-kontrolya-bezopasnosti-dorozhnogo-dvizheniya> (дата обращения: 19.01.2026)
4. *Борисов Е. А.* Применение БПЛА в кадастровой деятельности // Международный журнал гуманитарных и естественных наук. 2024. №7-1 (94). URL: <https://cyberleninka.ru/article/n/primenenie-bpla-v-kadastrovoy-deyatelnosti> (дата обращения: 19.01.2026)
5. *Ксенофонтов С. В.* Применение БПЛА при эксплуатации газораспределительных систем // Вестник магистратуры. 2021. №1-1 (112). URL: <https://cyberleninka.ru/article/n/primenenie-bpla-pri-ekspluatatsii-gazoraspredelitelnyh-sistem> (дата обращения: 19.01.2026)
6. *Орехов С. Ю., Бородулин А. В., Гайгеров М. А.* Исследование применения беспилотных летательных аппаратов и их основных элементов // StudNet. 2022. №6. URL: <https://cyberleninka.ru/article/n/issledovanie-primeneniya-bespilotnyh-letatelnyh-apparatov-i-ih-osnovnyh-elementov> (дата обращения: 19.01.2026)
7. Производство БПЛА в России [Электронный ресурс] // Tadviser. 2024-2026. URL: <https://www.tadviser.ru/a/842051> (дата обращения: 19.01.2026)
8. *Gamma Erich, Richard Helm, Ralph Johnson, John Vlissides.* Design patterns : elements of reusable object-oriented software [Electronic resource]. Anniversary edition. Boston : Addison-Wesley, 2018. 395 p.
9. *Баталин Р. Ю.* Проблемы адаптации информационных систем и программного обеспечения под ОС Astra Linux // Известия ТулГУ. Технические науки. 2024. №7. URL: <https://cyberleninka.ru/article/n/problemy-adaptatsii-informatsionnyh-sistem-i-programmnogo-obespecheniya-pod-os-astra-linux> (дата обращения: 19.01.2026).

АЛГОРИТМ АДАПТИВНОЙ БАЛАНСИРОВКИ ДЛЯ РАСПРЕДЕЛЁННЫХ СИСТЕМ С СИЛЬНО СКОШЕННОЙ НАГРУЗКОЙ

Бессолицын Антон Дмитриевич
МТУСИ, Магистрант, гр. М092401 (75)
besstoni2100@gmail.com

Гадасин Денис Вадимович
МТУСИ, заместитель заведующего кафедры СИТуС, к.т.н., доцент
dengadiplom@mail.ru

Галицкий Максим Викторович
МТУСИ, доцент кафедры СИТуС, Москва, Россия
m.v.galickiy@mtuci.ru

Гадасин Даниил Денисович
МТУСИ, магистрант группы М092501(70), Москва, Россия
gadasin115@gmail.com

Аннотация

В работе предложен и исследован алгоритм адаптивной балансировки нагрузки SLaB (Sketch-driven Locality-aware Adaptive Balancer), предназначенный для распределённых систем управления базами данных (РСУБД) и аналитических платформ, работающих в условиях сильно скошенной и быстро меняющейся нагрузки по ключам. Алгоритм сочетает двухконтурную архитектуру управления, вероятностные структуры данных (эскизы) и динамическое микрошардирование. Быстрый контур, работающий в миллисекундном масштабе, реализует маршрутизацию запросов по схеме power-of-two-choices с учётом локальности данных и текущей загрузки узлов, при этом не выполняя операций миграции. Медленный контур, работающий в масштабе секунд и минут, использует Count-Min Sketch, HyperLogLog, MinHash и Locality-Sensitive Hashing (LSH) для выделения горячих диапазонов ключей, построения микрошардов и принятия решений о миграции и репликации в рамках заданного бюджета на перемещение данных.

Ключевые слова:

мониторинг СУБД, проверка качества данных, разрешение конфликтов, асинхронная обработка, алгоритмы в СУБД.

Введение

Современные высоконагруженные сервисы – рекламные платформы, социальные сети, системы мониторинга, биржевые и платёжные системы – работают с потоками данных, которые характеризуются не только большим суммарным объёмом, но и крайне неравномерным распределением активности по ключам. Небольшое количество «горячих» сущностей, таких как популярные посты в социальных сетях, товары в период распродаж или метрики ключевых сервисов, может генерировать десятки и сотни тысяч запросов в секунду, в то время как подавляющее большинство ключей остаётся практически неактивным. На практике это приводит к выраженному Zipf-подобному распределению нагрузки, при котором небольшая доля ключей отвечает за значительную долю трафика.

Классические подходы к распределению данных в кластере, основанные на статическом шардировании с применением consistent hashing и виртуальных узлов, позволяют добиться формально равномерного распределения ключей по узлам. Однако этого оказывается недостаточно. Даже при идеальном равномерном распределении ключей с точки зрения пространства хешей реальная нагрузка остаётся скошенной: часть узлов оказывается регулярно перегруженной запросами к нескольким горячим ключам, в то время как другие ресурсы простаивают. В условиях жёстких требований к хвостам распределения задержек (p95, p99) подобные локальные перегрузки превращаются в серьёзную эксплуатационную проблему. Ситуацию усложняет динамический характер нагрузки. Всплески активности могут возникать вследствие внешних событий, маркетинговых кампаний, инцидентов в инфраструктуре. Для таких всплесков характерны как быстрый рост, так и относительно короткая продолжительность, что делает подходы с редкой периодической ребалансировкой принципиально

неэффективными: система успевает «захлебнуться» задолго до того, как алгоритм перераспределения данных отреагирует.

Дополнительной сложностью является работа с диапазонами и коррелированными ключами. В ряде приложений (например, аналитические запросы по временным рядам или событиям) значимыми являются не отдельные ключи, а группы ключей, используемые совместно. В этом случае важно, чтобы «логически близкие» данные были размещены таким образом, чтобы избежать избыточных межузловых обращений и обеспечивать локальность чтения. Указанные факторы формируют потребность в алгоритме, который, с одной стороны, способен реагировать на изменения нагрузки в очень коротких временных масштабах, а с другой – эффективно управляет размещением данных и реплик в более медленном режиме, ограничивая при этом объём миграций и сетевые накладные. Такой алгоритм должен опираться на компактные структуры для сбора статистики, избегать необходимости в централизованном тяжёлом координаторе и хорошо работать как с единичными горячими ключами, так и с диапазонами и кластерами коррелированных ключей. Ответом на эту потребность является предлагаемый в работе алгоритм SLaB.

1. Общий принцип работы распределённых СУБД и мотивация алгоритма SLaB

Типичная распределённая СУБД строится вокруг нескольких базовых идей, а ее схема представлена на рисунке 1.

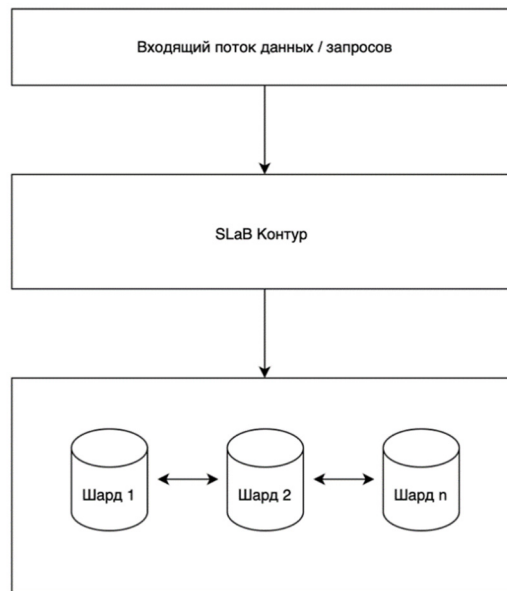


Рис. 1. Схема РСУБД с интегрированным SLaB контуром

Во-первых, пространство ключей или первичных идентификаторов данных делится на логические части – шарды [1]. Каждый шард отвечает за некоторый диапазон ключей или результатов хеш-функции. Во-вторых, для обеспечения отказоустойчивости и масштабируемости операций чтения каждый шард хранится в нескольких репликах на разных физических узлах. В-третьих, для направления запросов к нужному месту используется компонент маршрутизации: это может быть клиентская библиотека, прокси-слой либо встроенный механизм самой СУБД. В простейшем варианте маршрутизация осуществляется с помощью хеш-функции: по ключу вычисляется значение хеша, на его основе определяется шард, а далее – конкретный узел, на котором расположена одна из реплик шарда [2]. Для равномерности применяют множество виртуальных узлов: каждому физическому узлу соответствует несколько точек на кольце хеширования, что позволяет гибко подбирать распределение ключей. В случае выхода из строя узла или добавления нового выполняется процедура ребалансировки, при которой часть шардов переносится на другие узлы [3].

Однако на практике этого оказывается недостаточно. Во-первых, нагрузка по ключам крайне неравномерна. Даже если все шарды имеют примерно одинаковое количество ключей, интенсивность обращений к ним может различаться на порядок и более. Это приводит к возникновению «горячих» шардов и, как следствие, «горячих» физических узлов. Во-вторых, профиль нагрузки во времени оказывается нестабильным: ключ, который вчера был холодным, сегодня может стать одним из самых

горячих. Редкая периодическая ребалансировка не успевает отреагировать на такие изменения, и система длительное время работает в состоянии локальной перегрузки. Во многих системах используются механизмы селективной репликации горячих ключей: при обнаружении часто используемого ключа создаётся дополнительная реплика данных на другом узле, а маршрутизация начинает учитывать это [4]. Подход хорошо работает для отдельных ключей, но плохо масштабируется, когда горячими оказываются диапазоны или семейства ключей с высокой степенью корреляции по запросам. В этом случае необходимо управлять не отдельными точками, а структурой размещения диапазонов, что увеличивает сложность. Адаптивное микрошардирование, применяемое в ряде промышленных систем, позволяет уменьшать размер шардов и переносить только горячие части данных, оставляя холодные на месте. Такой подход более точен, но часто требует централизованного координатора, поддерживающего глобальное состояние системы и принимающего решения о переносах. В условиях очень больших кластеров это приводит к новым узким местам. Таким образом, типичная РСУБД нуждается в механизме, который, оставаясь достаточно простым и децентрализованным, способен: быстро реагировать на изменения нагрузки, учитывать локальность данных, ограничивать объём миграций и оперировать более мелкими единицами — микрошардами. SLaB строится именно вокруг этих требований.

2. Теоретический принцип работы алгоритма SLaB и иллюстративные схемы

Алгоритм SLaB реализует двухконтурную схему управления нагрузкой. Быстрый контур, обозначаемый как Контур А, работает в масштабе отдельных запросов и задач маршрутизации. Медленный контур, Контур В, функционирует в масштабе секунд и минут и отвечает за анализ накопленной статистики, выделение горячих микрошардов и планирование миграций и репликаций [5-7]. Работа быстрого контура можно описать следующим образом. При поступлении запроса с ключом k система определяет логический микрошард, которому принадлежит этот ключ. Микрошарды являются более мелкими единицами, чем традиционные шарды, их количество существенно превышает число узлов в кластере. Для найденного микрошарда извлекается список узлов, на которых расположены его реплики [8]. Из этого списка случайным образом выбираются два кандидата. Далее для каждого из них вычисляется интегральная метрика, отражающая текущую очередь запросов, загрузку CPU и локальность данных с точки зрения запросов к похожим ключам. Метрика может быть представлена линейной комбинацией вида 1

$$\begin{aligned} \text{score}_i = w_1 \cdot \text{queue_len}_j + w_2 \cdot \text{cpu}_j + \\ + w_3 \cdot (1 - \text{locality_bonus}_j), \end{aligned} \quad (1)$$

где коэффициенты w_1 , w_2 , w_3 подбираются экспериментально, а locality_bonus_j отражает наличие на узле «похожих» горячих ключей, выявленных делением MinHash на LSH. Запрос отправляется на узел с меньшим значением score . Такой подход является развитием идеи power-of-two-choices: достаточно сравнивать две случайно выбранные реплики, чтобы получить близкую к оптимальной балансировку нагрузки при минимальных накладных. Медленный контур работает иначе. Узлы кластера с некоторой периодичностью, как правило раз в несколько секунд, отправляют Контур В сжатые статистические данные в виде вероятностных эскизов и агрегированных метрик [9-12]. Используются Count-Min Sketch для оценок частоты обращений к ключам, HyperLogLog для оценки числа уникальных ключей за окно, MinHash и LSH для выявления групп ключей, которые часто встречаются вместе или демонстрируют похожий контекст использования. Объём этой телеметрии невелик и измеряется несколькими мегабайтами на узел при потоках в миллионы запросов в секунду [13]. На основании полученных эскизов Контур В вычисляет частоты обращений к микрошардам и определяет, какие из них являются «горячими». Вводится пороговое условие вида 2.

$$f_s > \gamma \cdot \left(\frac{Q_{\text{total}}}{M} \right), \quad (2)$$

где f_s – оценка частоты для микрошарда s , Q_{total} – общее число запросов в рассматриваемом окне, M – количество микрошардов, а γ – коэффициент, определяющий, насколько «горячим» должен быть микрошард. При обнаружении горячих микрошардов Контур В принимает решения о сплите, то есть о разбиении логического диапазона на более мелкие части, и о создании дополнительных реплик на менее загруженных узлах [14]. При этом учитывается заданный бюджет миграций: суммарный объём перемещаемых данных не должен превышать заранее установленную величину в единицу времени. Для иллюстрации работы SLaB удобно использовать рисунок 2.

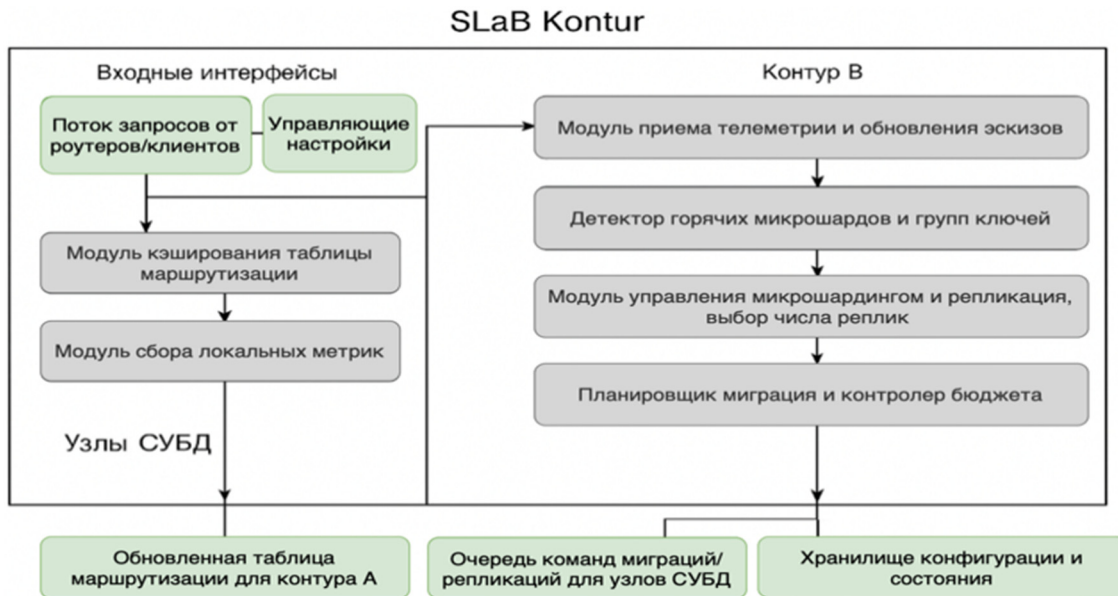


Рис. 2. Архитектура контура SLAB

3. Детальное описание архитектуры алгоритма и реализация

Архитектура SLaB реализуется в виде двух компонентов, логически соответствующих Контур А и Контур В. При этом оба компонента размещаются максимально близко к самим узлам СУБД, чтобы избежать лишних сетевых задержек. Контур А реализован как лёгкий роутер запросов (рис. 2). Он может быть оформлен как отдельный процесс или библиотека, встроенная в клиентский слой или прокси. В прототипе для ClickHouse был использован подход с дополнительной функцией `slab_route()` и внешним словарём, в котором хранится информация о маршрутизации микрошардов [15-16]. При поступлении запроса роутер извлекает ключ, находит соответствующий микрошард и список реплик, затем выбирает две случайные реплики и для каждой вычисляет `score` на основе локального состояния узла. Значения очереди и загрузки CPU передаются по телеметрии из самих узлов БД. `Locality_bonus` определяется на основании информации о кластерах ключей, полученной от Контура В. Далее запрос направляется на узел с лучшим `score`.

Важной особенностью является то, что этот выбор не требует глобального согласованного состояния: достаточно знать локальные метрики по нескольким узлам и актуальные правила маршрутизации. Контур В реализован как отдельный сервис. Он работает в режиме `side-car` либо в виде отдельного сервиса, доступного всем узлам кластера. Узлы БД с заданной периодичностью отправляют ему сжатую статистику: `Count-Min Sketch` по наблюдаемым ключам, `HyperLogLog` для оценки числа активных ключей, векторы `MinHash` и агрегированные показатели нагрузки [17].

Контур В агрегирует эти данные, вычисляет частоты для микрошардов, определяет горячие и холодные области, и формирует решения о сплите и миграциях. Процесс сплита микрошарда заключается в том, что исходный диапазон ключей делится на два или более поддиапазонов. Для каждого нового микрошарда определяется набор узлов-кандидатов для размещения реплик. Контур В учитывает текущую нагрузку узлов, доступный объём памяти и дисков, а также заданный бюджет миграций. На этой основе выбирается набор операций: какие микрошарды перенести, где создать дополнительные реплики, какие реплики можно удалить.

Для каждого действия оценивается ожидаемое снижение максимальной нагрузки и «стоимость» в виде объёма данных, который потребуется перенести. Выбирается подмножество действий, дающее максимальный эффект при ограничении на бюджет. После этого Контур В передаёт на узлы команды, реализующие миграции, и обновляет словарь маршрутизации, который использует Контур А. Для того чтобы миграции не мешали боевому трафику, переносы данных выполняются «тонкой струей» – ограничивается скорость передачи, например 50-100 МБ/с на узел [18-19]. Это напоминает подход `trickle-replication`, применяемый в ряде систем типа `Cassandra`: репликация и миграции происходят достаточно медленно, чтобы не нарушать SLA по основным операциям, но при этом система в целом успевает адаптироваться к поменявшейся нагрузке.

4. Модели оценки, нагрузочные сценарии и экспериментальные результаты

В качестве отдельного этапа работы над алгоритмом планируется проведение комплексной оценки эффективности алгоритма SLaB. Целью этого этапа является количественное подтверждение того, что предложенный подход действительно улучшает характеристики распределённой СУБД по сравнению с базовыми решениями, основанными на статическом шардировании и периодической ребалансировке. В рамках планируемых экспериментов предполагается измерять несколько групп метрик. С одной стороны, будут анализироваться характеристики латентности: медиана, 95-й и 99-й процентиль времени ответа на запросы при различных сценариях нагрузки. С другой стороны, будет оцениваться суммарный объём миграций данных за заданный интервал времени, что позволит судить о «агрессивности» алгоритма с точки зрения переноса данных.

Дополнительно планируется оценивать энергопотребление кластера и разброс загрузки CPU по узлам, поскольку выравнивание нагрузки должно давать заметный эффект на эффективность использования аппаратных ресурсов. Для проведения такого анализа предполагается использовать кластер из 128 машин с типичной для промышленных систем конфигурацией: многопроцессорные узлы с большим количеством ядер, значительными объёмами оперативной памяти и быстрыми NVMe-накопителями, соединённые высокоскоростной сетью. На этом стенде будет воспроизведено несколько сценариев нагрузки. Один из сценариев будет основан на YCSB с Zipf-подобным распределением обращений к ключам (например, с параметром $\theta \approx 0,99$), поверх которого планируется моделировать искусственные всплески активности на малой доле ключей [20-22]. Такой сценарий позволит проверить устойчивость SLaB к крайне скошенной и «неровной» нагрузке. Другой сценарий будет ориентирован на аналитические запросы по временным диапазонам и может быть реализован на основе бенчмарка TPC-DS с искусственным перекосом по дате, что имитирует частые запросы к последним периодам. Отдельный интерес представляет сценарий, основанный на воспроизведении реального продакшн-трафика (после анонимизации и тщательной очистки), который позволит оценить алгоритм в условиях, максимально приближенных к работе промышленной системы. Сравнение планируется проводить между несколькими конфигурациями. В качестве базовой конфигурации будет использовано статическое распределение данных по многочисленным виртуальным узлам, типичным для современных распределённых СУБД.

Также в план оценки входит режим работы систем без дополнительной балансировки (например, конфигурация по умолчанию в ClickHouse или Cassandra с виртуальными узлами) [23-31]. Поверх таких конфигураций будет включаться алгоритм SLaB, интегрированный в путь обработки запросов и в контур миграций данных, что позволит сопоставить результаты «до» и «после». Ожидается, что по итогам таких экспериментов SLaB продемонстрирует существенное улучшение хвостов распределения задержек. Интуитивно, за счёт сочетания локальной маршрутизации по двум кандидатам и глобальной адаптации структуры микрошардов алгоритм должен снижать 99-й процентиль латентности в несколько раз по сравнению с базовым статическим распределением, при этом медианные задержки останутся на уровне базовых конфигураций, а основное преимущество проявится именно в уменьшении «тяжёлых» хвостов. С точки зрения миграций предполагается, что SLaB будет переносить существенно меньший объём данных, чем традиционные схемы периодической ребалансировки, поскольку алгоритм работает на уровне микрошардов и оперирует только выделенными горячими диапазонами, а не крупными статическими шардами [26]. За счёт выравнивания нагрузки и сокращения числа перегруженных узлов ожидается снижение разброса по загрузке CPU и, как следствие, возможность перевода части узлов в энергосберегающие режимы без потери общей пропускной способности. Таким образом, ожидаемые результаты включают не только улучшение латентности, но и уменьшение объёма миграций и повышение энергоэффективности, что делает применение SLaB перспективным в промышленных кластерах.

Заключение

В работе был предложен и рассмотрен алгоритм SLaB, предназначенный для балансировки нагрузки в распределённых СУБД и аналитических системах, работающих в условиях сильно скошенной и динамически меняющейся нагрузки. Алгоритм строится вокруг двухконтурной архитектуры: быстрый Контур А реализует маршрутизацию запросов на основе локальных метрик и схемы power-of-two-choices, а медленный Контур В собирает вероятностные эскизы, выделяет горячие микрошарды, планирует их сплит, миграции и репликацию, соблюдая заданный бюджет миграций. В качестве базовых строительных блоков используются Count-Min Sketch, HyperLogLog, MinHash и LSH, что позволяет ограничить объём телеметрии несколькими мегабайтами на узел даже при нагрузках в миллионы запросов в секунду.

Математическая модель алгоритма показывает, что предложенный подход позволяет приближаться к оптимальной балансировке при ограниченной информации, а явное бюджетирование миграций удерживает объём переносимых данных под контролем. Планируемая экспериментальная оценка на кластере из 128 узлов ClickHouse должна подтвердить ожидаемое снижение хвостов распределения задержек, уменьшение объёма миграций и снижение энергопотребления кластера без необходимости использования тяжёлого централизованного координатора.

Направления дальнейших исследований включают интеграцию методов прогнозирования нагрузки на основе временных рядов и градиентного бустинга, применение обучения с подкреплением для автоматической настройки параметров алгоритма, а также адаптацию SLaB к другим типам РСУБД и сценариям нагрузки, включая гибридные транзакционно-аналитические нагрузки. Всё это делает SLaB перспективным инструментом для построения устойчивых и энергоэффективных распределённых хранилищ данных следующего поколения.

Литература

1. *Гадасин Д. В., Шведов А. В., Пантелева К. А., Гадасин Д. Д.* Определение порога количества информации для возможности структурирования данных // Телекоммуникационные и вычислительные системы : Юбилейный сборник трудов тридцатого международного научно-технического форума, Москва, 12-15 декабря 2022 года. Москва: Издательство МБА, 2022. С. 125-130. EDN MYMHP.
2. *Яковенко Н. В., Гадасин Д. В., Коцич Л.* Повышение точности коэффициента влияния ошибок в информационных системах с применением метода обратного распространения ошибки // Системы синхронизации, формирования и обработки сигналов. 2024. Т. 15, № 4. С. 35-42. EDN CMFVNH.
3. *Аль Мусави О.А.Р., Кравец О.Я.* Исследование алгоритмов повторной оптимизации запросов в облачных базах данных // Решение. 2022. Т. 1. С. 168-171.
4. *Gadassin D. V., Melnikova E. M., Palibza S. A., Gadassin D. D.* An Algorithm for Distributing Data between Storage Systems Based on the Property of Self-Similarity // 2024 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2024, pp. 1-7, doi: 10.1109/IEEECONF60226.2024.10496761.
5. *Salloum S. et al.* Big data analytics on Apache Spark // International Journal of Data Science and Analytics. 2016. Т. 1. С. 145-164.
6. *Кузнецов С. Д.* Основы современных баз данных. Информационно-аналитические материалы: [Электрон.ресурс] Режим доступа: <http://www.citforum.ru/database/osbd/contents.shtml>. Дата доступа: 20.12.2024.
7. *Роб Питер, Коронелл К.* Системы баз данных: проектирование, реализация и управление : Пер. с англ. 5-е изд. СПб. : БХВ-Петербург, 2004. 1024 с.
8. *Gadassin D. V., Shvedov A. V., Kuzin I. A.* Reconstruction of a Three-Dimensional Scene from its Projections in Computer Vision Systems // 2021 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex, TIRVED 2021 – Conference Proceedings, Moscow, 11-12 ноября 2021 года. Moscow, 2021. DOI 10.1109/TIRVED53476.2021.9639161. EDN CKSNPA.
9. *Гадасин Д. В., Пак Е. В., Коровушкина В. М., Мелькова Е. К.* Предобработка текстовой информации на основе термов естественного языка // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12, № 1. С. 4-11. EDN PDGAVP.
10. *Gadassin D. V., Shvedov A. V., Klygina O. G.* Organization of Interaction Between the Concept of Fog Computing and Segment Routing for the Provision of IoT Services in Smart Grid Networks // Wave Electronics and Its Application in Information and Telecommunication Systems. 2022. Vol. 5, No. 1, pp. 141-146. EDN UQSHRH.
11. *Золотарева П. Ю., Гадасин Д. В., Маклачков К. А.* Методы обработки информации в распределённых информационных системах // Тенденции развития Интернет и цифровой экономики : Труды VI Международной научно-практической конференции, Симферополь-Алушта, 01-03 июня 2023 года. Симферополь: ИП Зуева, 2023. С. 187-189. EDN LGONZK.
12. *Гадасин Д. В.* Построение бинарного дерева минимальной цены // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 11. С. 38-44. DOI 10.36724/2072-8735-2024-18-11-38-44. EDN GMCEWG.
13. *Shevelev S. V., Shvedov A. V., Gadassin D. V., Vakurin I. S.* Syntax and probability vectors in search query // Wave Electronics and Its Application in Information and Telecommunication Systems. 2023. Vol. 6, No. 1, pp. 407-414. EDN TVFKOH.
14. *Гадасин Д. В., Шведов А. В.* Проблемы интеграции концепции "Интернет вещей" и облачных вычислений // Технологии информационного общества : Материалы XIII Международной отраслевой научно-технической конференции, Москва, 20–21 марта 2019 года. Т. 2. М.: Издательский дом Медиа Паблишер, 2019. С. 22-23. EDN MEQRFA.
15. *Gadassin D. V., Shvedov A. V., Vakurin I. S.* Determination of Semantic Proximity of Natural Language Terms for Subsequent Neural Network Training // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744290. EDN LASMDY.

16. *Гадасин Д. В., Шведов А. В., Мелькова Е. К.* Структурирование данных исходя из центра масс // Актуальные проблемы и перспективы развития экономики : Труды XXI Международной научно-практической конференции, Симферополь-Гурзуф, 20-22 октября 2022 года. Симферополь: Крымский федеральный университет им. В.И. Вернадского, 2022. С. 266-268. EDN RFCCST.
17. *Gadasin D. V., Shvedov A. V., Yudin A. A.* Clustering methods in large-scale systems // *Synchroinfo Journal*. 2020. Vol. 6, No. 5, pp. 21-24. DOI 10.36724/2664-066x-2020-6-5-21-24. EDN XHNSYV.
18. *Gadasin D. V., Shvedov A. V., Kuzin I. A.* A model for representing the color and depth metric characteristics of objects in an image // 2021 Systems of Signal Synchronization, Generating and Processing in Telecommunications, SYNCHROINFO 2021 – Conference Proceedings, Svetlogorsk, Kaliningrad Region, 30 июня – 02 июля 2021 года. Svetlogorsk, Kaliningrad Region, 2021. P. 9488349. DOI 10.1109/SYNCHROINFO51390.2021.9488349. EDN YAYZVP.
19. *Zolotukhin P. A., Melkova E. K., Gadasin D. V., Korovushkina V. M.* Using Intelligent Testing as a Tool to Improve the Quality of Transport Information Systems // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744348. EDN NOMJLX.
20. *Гадасин Д. В., Шведов А. В.* Применение транспортной задачи для балансировки нагрузки в условиях нечеткости исходных данных // *T-Comm: Телекоммуникации и транспорт*. 2024. Т. 18, № 1. С. 13-20. DOI 10.36724/2072-8735-2024-18-1-13-20. EDN WKNPIX.
21. *Гадасин Д. В., Шведов А. В., Алексеева Е. А.* Информационная энтропия в стохастических сетях связи // Телекоммуникационные и вычислительные системы 2020 : Труды международной научно-технической конференции, Москва, 14-17 декабря 2020 года / Московский технический университет связи и информатики. Москва: Научно-техническое издательство "Горячая линия-Телеком", 2020. С. 108-116. EDN IOGLQH.
22. *Shvedov A. V., Gadasin D. V., Pak E. V.* Application of the Backman Model for the Distribution of Traffic Flows in Networks with Segment Routing // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744344. EDN RBMTBQ.
23. *Gadasin D. V., Koltsova A. V., Gadasin D. D.* Algorithm for Building a Cluster for Implementing the 'Memory as a Service' Service in the IoT Concept // 2021 Systems of Signals Generating and Processing in the Field of on Board Communications, Conference Proceedings, Moscow, 16-18 марта 2021 года. Moscow, 2021. P. 9416112. DOI 10.1109/IEEECONF51389.2021.9416112. EDN VRPCFG.
24. *Гадасин Д. В., Смальков Н. А., Кузин И. А.* Использование метода роя частиц для балансировки нагрузки в сетях Интернета вещей // Системы синхронизации, формирования и обработки сигналов. 2022. Т. 13, № 2. С. 17-23. EDN LIUWNT.
25. *Gadasin D. V., Shvedov A. V., Koltsova A. V.* Cluster model for edge computing // 2020 International Conference on Engineering Management of Communication and Technology, EMCTECH 2020 : Proceedings, Vienna, 20–22 октября 2020 года. New York: Institute of Electrical and Electronics Engineers Inc., 2020. P. 9261538. DOI 10.1109/EMCTECH49634.2020.9261538. – EDN FGDLSA.
26. *Шведов А. В., Гадасин Д. В., Коровушкина В. М., Мелькова Е. К.* Интеллектуальное тестирование как способ повышения качества информационной системы // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12, № 2. С. 43-52. EDN GOLZGE.
27. *Гадасин Д. В., Шведов А. В., Кузин И. А.* Трехмерная реконструкция объекта по одному изображению с использованием глубоких сверточных нейронных сетей // *T-Comm: Телекоммуникации и транспорт*. 2022. Т. 16, № 7. С. 29-35. DOI 10.36724/2072-8735-2022-16-7-29-35. EDN YTLCNW.
28. *Shvedov A. V., Gadasin D. V., Alyoshintsev A. V.* Segment routing in data transmission networks // *T-Comm: Телекоммуникации и транспорт*. 2022. Vol. 16, No. 5, pp. 56-62. DOI 10.36724/2072-8735-2022-16-5-56-62. EDN VAYLJQ.
29. *Alyoshintsev A. V., Gadasin D. V., Vakurin D. S., Chelyshkov P. D.* Methods for evaluating the noise immunity of modems // *T-Comm: Телекоммуникации и транспорт*. 2025. Vol. 19, No. 9, pp. 50-58. DOI 10.36724/2072-8735-2025-19-9-50-58. EDN TGKCQD.
30. *Гадасин Д. В.* Способ определения основных узлов сети для анализа ее состояния // *T-Comm: Телекоммуникации и транспорт*. 2025. Т. 19, № 12. С. 16-24. DOI 10.36724/2072-8735-2025-19-12-16-24. EDN FGAATI.
31. *Мелькова Е. К., Шведов А. В., Трemasова Л. А., Гадасин Д. В.* Организация кластера исходя из функции принадлежности // Системы синхронизации, формирования и обработки сигналов. 2023. Т. 14, № 1. С. 30-39. EDN CNVIJU.

ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ ИНСТРУМЕНТОВ В ПРОЦЕССЕ РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ ПОТОКАМИ И ПРОИЗВОДСТВЕННЫМИ ПРОЦЕССАМИ ТКАЦКОЙ ФАБРИКИ

Караваяева Эвелина Андреевна

*Московский технический университет связи и информатики,
студент группы БСТ2303, Москва, Россия,
karavaeva1921@mail.ru*

Ковтун Игорь Иванович

*Московский технический университет связи и информатики,
к.т.н., доцент, доцент кафедры «Системное программирование», Москва, Россия,
i.i.kovtun@mtuci.ru*

Аннотация

Представлены научные и практические результаты, полученные в ходе разработки автоматизированной системы управления информационными и производственными потоками ткацкой фабрики «Ивановский трикотаж». Используются формальные методы функционального моделирования IDEF0, реляционного проектирования баз данных. Практическая интеграция формальных методов и моделей осуществлена на языке SQL посредством применения методов искусственного интеллекта – заполнения слотов заранее predetermined SQL-фреймов, полученных по результатам анализа функциональной модели. Разработанный прототип позволяет отслеживать материальные потоки и оптимизировать документооборот, что способствует повышению эффективности и прозрачности управления производственными процессами ткацкой фабрики.

Ключевые слова

ткацкая фабрика, комплексная автоматизация, формализация информационных и производственных процессов, реляционная модель данных, функциональная модель IDEF0, Structured Query Language, SQL-фрейм

Введение

Лёгкая промышленность является одной из базовых отраслей экономики Российской Федерации, обеспечивающей производство массовых потребительских товаров и устойчивость региональных производственных цепочек [1]. В структуре отрасли ткацкие предприятия занимают ключевое место, а повышение эффективности их функционирования связано с задачами рационального использования ресурсов, снижения издержек и обеспечения управляемости производственных процессов.

В современных условиях эффективное управление деятельностью промышленного предприятия невозможно без применения средств автоматизации, компьютеризации, интеллектуализации и роботизации. Рост объёмов обрабатываемой информации и усложнение производственно-логистических цепочек обуславливают необходимость внедрения автоматизированных систем управления, обеспечивающих согласованность процессов и поддержку управленческих решений [2]. Обоснованный выбор способов автоматизации требует предварительного формализованного описания текущих и целевых процессов предприятия на основе моделей AS-IS и TO-BE [3].

В то же время, использование отдельных моделей имеет ограничения – реляционные модели формализуют структуру данных, но не отражают логику процессов, тогда как функциональные модели, в частности IDEF0, описывают процессы, не обеспечивая, при этом, полного представления о данных. В связи с вышеуказанным, целью работы, итоги которой кратко доложены в настоящей статье, является разработка и обоснование подхода к интеграции функциональной и реляционной моделей в процессе проектировании автоматизированной системы управления ткацкой фабрикой. Для достижения цели решаются задачи анализа бизнес-процессов, построения функциональных моделей, разработки логической и даталогической моделей данных и их практической интеграции [3, 4].

1. Функциональное моделирование процессов ткацкой фабрики

Функционирование ткацкой фабрики планируется с учетом высокой степени координации между этапами приема и обработки заказов, управления запасами, складским учетом и отправкой готовой

продукции. Торгово-складские и логистические процессы играют ключевую роль в эффективности деятельности предприятия, поскольку именно они гарантируют бесперебойность производственного цикла и своевременную реализацию обязательств по отношению к заказчикам. Исходя из этого, анализ функциональной структуры предприятия целесообразно начать с изучения перечисленных процессов.

В качестве инструмента для структурированного описания бизнес-процессов предприятия выбрана методология функционального моделирования IDEF0. В ходе первого этапа проведен анализ существующей структуры торгово-складской и логистической деятельности ткацкой фабрики по принципу AS-IS, отражающий реальный порядок выполнения операций, включая обработку заказов, учет материальных ресурсов и управление запасами. Пример функциональной декомпозиции представлен на рис. 1.

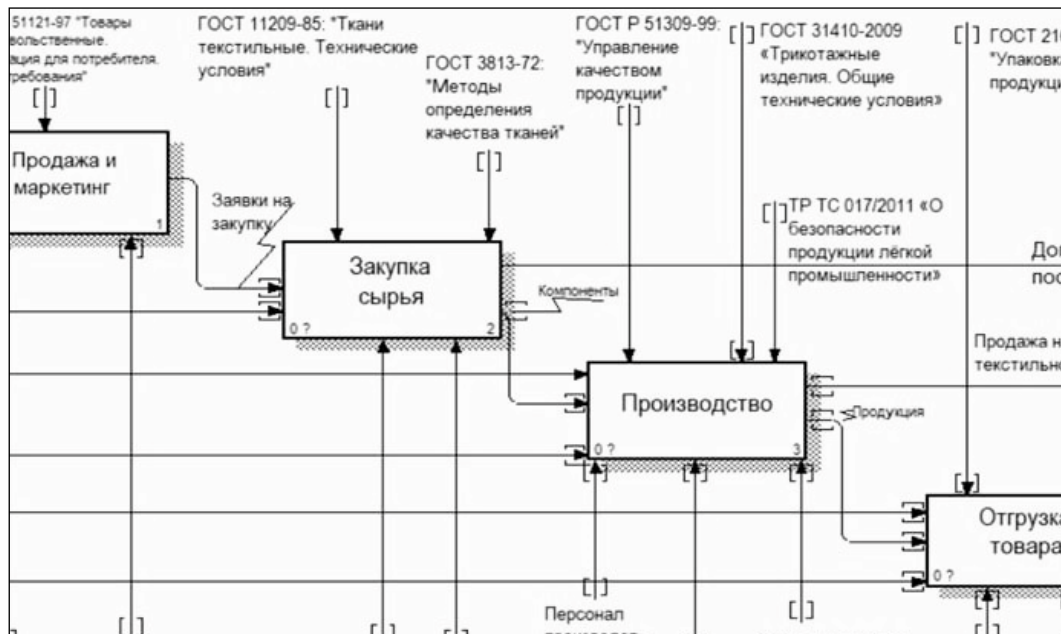


Рис. 1. Фрагмент декомпозиция процесса торгово-складской и логистической деятельности

Анализ представленной функциональной модели позволяет выделить общие группы процессов, определяющих деятельность ткацкой фабрики, включая приём и обработку заказов, управление материальными ресурсами, складской учёт и организацию отгрузки готовой продукции. Установлено, что указанные процессы характеризуются высокой степенью информационной взаимосвязанности и требуют постоянного обмена актуальными данными между различными подразделениями предприятия.

Существенной особенностью выявленной функциональной структуры является то, что значительная часть управленческих решений формируется на основе информации, поступающей из разнородных источников и обрабатываемой различными исполнителями. При отсутствии единого информационного пространства такое обстоятельство приводит к увеличению времени согласования действий, снижению оперативности управления и повышению риска возникновения ошибок, связанных с несвоевременным обновлением данных.

Проведённый анализ показал, что наибольшее влияние на непрерывность производственного цикла и выполнение обязательств перед заказчиками оказывают торгово-складские и логистические процессы. В связи с указанным фактором для последующего детального исследования и формализации в целях проектирования и создания автоматизированной системы управления выбран процесс отгрузки готовой продукции как один из наиболее критичных с точки зрения координации материальных и информационных потоков.

Функциональная модель процесса отгрузки построена также в нотации IDEF0 и отражает последовательность операций, выполняемых после завершения производства готовой продукции и её размещения на складе. В модели зафиксированы основные этапы, включающие формирование заявки на отгрузку, подготовку продукции к передаче, оформление сопроводительной и транспортной документации, организацию транспортировки и передачу продукции клиенту. Соответствующая декомпозиция процесса представлена на рис. 2.



Рис. 2. Фрагмент декомпозиции отгрузки товара

Анализ разработанной модели показывает, что процедура отправки товара осуществляется в строгом соответствии с товарно-транспортной накладной, которая оформляется непосредственно после доставки продукции на склад. Ключевыми данными для начала процесса служат: клиентский заказ, информация об остатках готовой продукции на складских площадях и действующие правила оформления сопроводительной документации. Итогом выполнения процесса является отгруженная партия товаров, заверенные сопроводительные документы и факт подтверждения выполнения обязательств перед клиентом.

Особое внимание в модели уделено этапу определения логистических параметров. Объем отправляемой продукции прямо пропорционален весу и объему груза, что оказывает влияние на выбор транспортного средства, разработку оптимального маршрута и установление сроков доставки. На данном этапе происходит взаимодействие с логистическими компаниями, формирование запросов на перевозку и подготовка полного пакета документации, необходимой для юридически грамотной передачи груза конечному потребителю.

Исследование процесса поставки позволяет выделить критические аспекты, касающиеся оформления документации, координирования транспортировки и передачи товара заказчику. Полученные данные учитываются при формировании требований к автоматизированной системе управления, включая структуру данных, алгоритмы учета отгрузок и поддержку логистических операций.

В рамках функционального моделирования согласно методологии IDEF0 для каждого выделенного процесса определены входные данные, управляющие воздействия, ресурсы и исходящие результаты. В качестве входных данных для процессов выступают клиентские заказы, сведения о количестве доступных товарных единиц и нормативные требования к документам. Управляющими воздействиями являются регламенты производственной и складской деятельности, договорные обязательства и внутренние распоряжения предприятия. Ресурсами для выполнения процессов выступают сотрудники подразделений, складская инфраструктура и информационные системы компании. Итогом являются отгруженные товары, надлежаще оформленная документация и подтверждения исполнения клиентских заказов.

2. Формализация информационных потоков и разработка реляционной модели данных

Применение функционального моделирования к информационным и производственным процессам ткацкой фабрики позволяет установить иерархию и взаимодействие выполняемых операций. Однако для создания автоматизированной системы управления необходима формализованная структура информации, используемой в процессе управления. В условиях сложного производственного процесса и большого количества обрабатываемых данных, отсутствие четкого представления информационных потоков затрудняет анализ текущего состояния ключевых показателей предприятия и снижает эффективность принимаемых решений. В связи с этим, для формализации информационных потоков

предложена реляционная модель данных, направленная на поддержку основных процессов управления ткацким производством [5].

Анализ функциональной модели IDEF0 показал, что в работе ткацкой фабрики возникают различные информационные потоки, связанные с обработкой заказов клиентов, учетом сырья и материалов, планированием производственного процесса, управлением складскими операциями, а также отгрузкой готовой продукции. Для организации таких потоков выделены основные информационные элементы предметной области и логические связи между ними. Результатом такого анализа явилась логическая модель данных, отражающая структуру информационного обеспечения управления предприятием, см. рис. 3.



Рис. 3. Фрагмент логической модели данных информационных потоков ткацкой фабрики

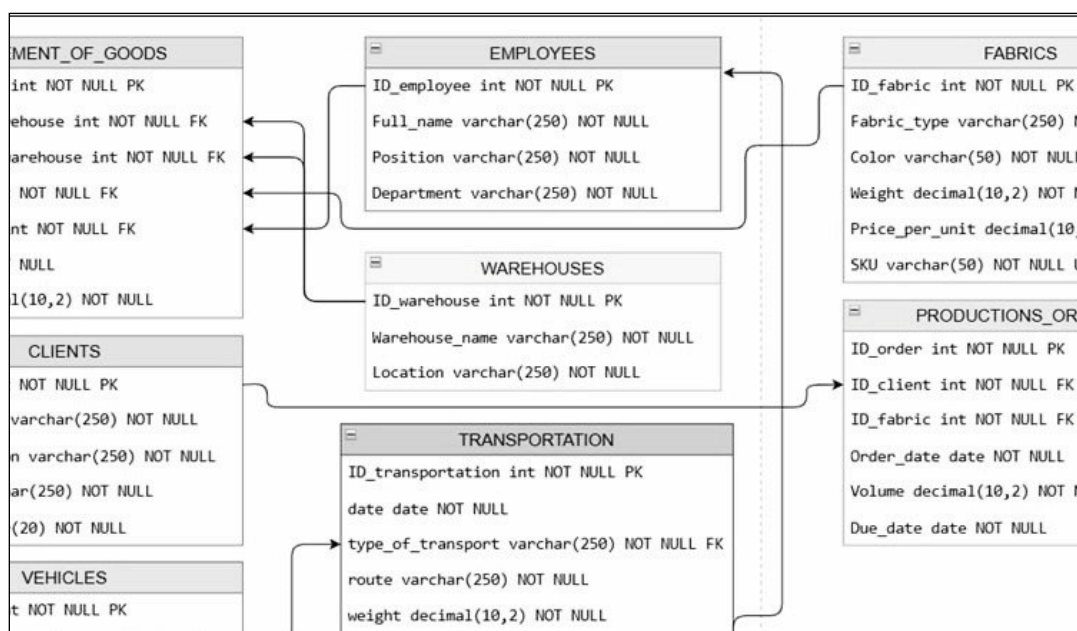


Рис. 4. Фрагмент даталогической модели данных автоматизированной системы управления ткацкой фабрики

На основе построенной логической модели разработана даталогическая модель (см. рис. 4), которая детализирует структуру хранения данных в терминах реляционной СУБД. На данном уровне определен состав таблиц, их атрибуты, первичные и внешние ключи, а также ограничения целостности, обеспечивающие согласованность и непротиворечивость данных. Разработанная структура данных нацелена на отражение логики функционирования процессов предприятия и исключение излишнего дублирования информации.

Результаты проектирования реляционной модели данных представлены в виде структур ключевых таблиц базы данных, отражающих основные объекты управления ткацкой фабрики. Примером таких таблиц являются таблицы Clients и Fabrics, см. таблицы 1, 2 соответственно. В указанных таблицах формализованы сведения о клиентах, заказах, номенклатуре продукции и материальных ресурсах, что создаёт основу для дальнейшей реализации аналитических и управленческих функций проектируемой системы.

Таблица 1

Структура таблицы Clients

Название поля	Тип данных	Пояснение	Ключи
Id	Int	Первичный ключ	Primary key
Company_name	Varchar	Название компании	Alternative key
Contact_person	Varchar	Контактное лицо	
Address	Varchar	Адрес	
Number	Varchar	Номер телефона	

Таблица 2

Структура таблицы Fabrics

Название поля	Тип данных	Пояснение	Ключи
Id	Int	Первичный ключ	Primary key
Fabric_type	Varchar	Тип ткани	
Color	Varchar	Цвет	
Weight	Decimal	Масса (вес)	
Unit_price	Decimal	Цена за единицу	
Article	Varchar	Артикул	Alternative key

Представленные структуры таблиц фиксируют результат формализации информационных потоков и отражают согласованность проектируемой базы данных с функциональной моделью процессов предприятия. Полученная реляционная модель служит информационной основой для последующей интеграции процессного и информационного представлений в рамках автоматизированной системы управления.

3. Интеграция функциональной и реляционной модели в процессе описания системы управления ткацкой фабрикой

Объединение функционального и реляционного подходов нацелено на гармонизацию представления процессов и информации в деятельности ткацкой фабрики в контексте автоматизированной системы управления. В отличие от отдельного применения указанных моделей, интегрированный подход позволяет установить взаимосвязь между исполняемыми задачами управления и конкретными информационными элементами, а также процедурами обработки данных, тем самым повышая понятность управленческих действий и обоснованность принимаемых решений.

В представленной работе интеграция достигается посредством сопоставления функций, определённых в IDEF0-модели, с соответствующими объектами и отношениями реляционной модели данных. Подобное сопоставление гарантирует чёткую связь между процессами предприятия и используемой в их реализации информацией, и также позволяет формализованно определить требования к обработке данных в среде автоматизированной системы управления.

Практическое осуществление интеграции реализовано, основываясь на логике бизнес-процессов предприятия, с использованием языка Structured Query Language, SQL, который использован как средство формализации операций доступа и манипулирования данными [6]. Сначала, см. Листинг 1, происходит разработка таблиц базы данных и определение связей между ними, отображающих структуру

предметной области, поскольку объединение функционального и реляционного моделирования в рамках автоматизированной системы управления предполагает не только согласование архитектуры данных и процессов, но и формализованный подход к управленческим схемам.

Листинг 1

Фрагмент SQL-кода создания таблиц базы данных

```
CREATE TABLE FABRICS (
  ID_fabric INT PRIMARY KEY,
  Fabric_type VARCHAR(250) NOT NULL,
  Color VARCHAR(50) NOT NULL,
  Weight DECIMAL(10, 2) NOT NULL,
  Price_per_unit DECIMAL(10, 2) NOT NULL,
  SKU VARCHAR(50) NOT NULL UNIQUE
) TABLESPACE WAREHOUSE;

CREATE TABLE CLIENTS (
  ID_client INT PRIMARY KEY,
  Company_name VARCHAR(250) NOT NULL,
  Contact_person VARCHAR(250) NOT NULL,
  Address VARCHAR(250) NOT NULL,
  Phone VARCHAR(20) NOT NULL
) TABLESPACE COMPANY;
```

В данном исследовании язык SQL рассматривается не просто как инструмент реализации базы данных, но и как формальный инструмент для описания стандартных управленческих воздействий, соответствующих функциям, выявленным в IDEF0-модели, что позволило разработать аналитические запросы, пример представлен в Листинге 2, ориентированные на решение типовых управленческих задач. При этом, использованы классические методы искусственного интеллекта – заранее предопределенные слоты SQL-фреймов, полученных по результатам анализа функциональной модели, заполняются данными, полученными в ходе анализа реляционной модели [7]. Такие запросы, соотносятся с функциями управления, определёнными в функциональной модели, и позволяют получать агрегированные показатели, характеризующие состояние производства, выполнение заказов и обеспеченность ресурсами.

Листинг 2

Пример аналитического SQL-запроса для поддержки управленческих решений

```
Найти пары складов с наибольшим объемом перемещений между ними:
SELECT w1.name_warehouse as from_warehouse,
       w2.name_warehouse as to_warehouse,
       COUNT(*) as movements_count,
       SUM(m.volume) as total_volume
FROM movement m
JOIN warehouse w1 ON m.sender = w1.id
JOIN warehouse w2 ON m.recipient = w2.id
GROUP BY w1.id, w2.id, w1.name_warehouse, w2.name_warehouse
ORDER BY movements_count DESC;
```

Данный запрос реализует функцию анализа движения товарно-материальных ценностей между складскими помещениями, которая выделена в рамках функциональной модели складского учета. Получаемые обобщенные показатели дают возможность оценить активность внутренних логистических операций и обнаружить потенциальные проблемные зоны в системе распределения товаров.

Применение SQL-запросов в рамках интегрированной модели позволяет придать формализованный вид стандартным управленческим задачам, таким как отслеживание исполнения заказов, анализ загруженности производственных ресурсов и наблюдение за уровнем запасов на складе. Построение аналитических выборок, опирающихся на согласованную структуру данных, уменьшает вероятность возникновения искажений в информации и гарантирует надежность результатов анализа.

С точки зрения методологии, объединение функциональной и реляционной моделей формирует базис для перехода от простого описания процессов и данных к поддержке принятия управленческих решений. Автоматизированная система управления в этом контексте рассматривается не только как средство ведения учета, но и как инструмент анализа текущего положения дел на предприятии и прогнозирования последствий принимаемых управленческих решений [8].

Заключение

В ходе проведенного исследования разработан и обоснован подход к интеграции функциональной и реляционной моделей в процессе проектирования автоматизированной системы управления информационными и производственными потоками ткацкой фабрики. В рамках работы выполнен анализ деятельности предприятия лёгкой промышленности, построена функциональная модель процессов в нотации IDEF0, а также разработаны логическая и даталогическая модели данных, отражающие структуру информационного обеспечения управления.

Показано, что использование функциональной и реляционной модели по отдельности не позволяет в полной мере отразить совокупность процессов и информационных взаимосвязей, характерных для ткацкого производства. Интеграция данных подходов обеспечивает согласованность процессного и информационного представлений, что повышает прозрачность управления и снижает риск принятия решений на основе неполной или недостоверной информации.

Практическая реализация интеграции с применением языка SQL продемонстрировала возможность формализации типовых управленческих и аналитических запросов, напрямую связанных с функциональной структурой предприятия. Это позволяет рассматривать автоматизированную систему управления не только как средство учёта, но и как инструмент анализа состояния производственных процессов и поддержки управленческих решений.

С методологической точки зрения полученные результаты демонстрируют возможность формализованного перехода от функционального описания производственных процессов к реляционному представлению информационных потоков с последующей реализацией управленческих сценариев в виде SQL-запросов. Предложенный подход позволяет рассматривать автоматизированную систему управления как интегрированную модель процессов и данных, а не как совокупность разрозненных программных решений.

Практическая ценность интегрированного функционально-реляционного подхода заключается в возможности его использования на этапе предпроектного обследования предприятия для обоснования требований к автоматизации, оценки реализуемости проектных решений и анализа последствий управленческих воздействий [9]. Это позволяет применять разработанную методику не только для ткацких фабрик, но и для других предприятий лёгкой промышленности с аналогичной структурой производственных и логистических процессов.

Литература

1. *Воронов С.В.* Современные проблемы автоматизации производственной логистики на малых предприятиях // Вестник экономики и промышленности. 2021. № 3. С. 45-49.
2. ГОСТ 31410–2009. Трикотажные изделия. Общие технические условия. М.: Стандартинформ, 2010.
3. *Ковтун И.И.* Функционально-реляционный анализ вариантов автоматизации сложных организационно-технических систем. Монография. СПб.: НИЦ АРТ, 2024. 214 с.
4. *Ковтун И.И.* Теория и практика проектирования государственных информационных систем. Учебное пособие. СПб.: НИЦ АРТ, 2023. 194 с.
5. *Полищук Ю.В.* Базы данных и их безопасность. Учебное пособие. М.: ИНФРА-М, 2025. 210 с.
6. *Ковтун И.И., Романенко Г.С.* Компьютерная реализация методов формального анализа сложных организационно-технических систем в процессе проведения комплексной автоматизации // Информатизация и связь. 2022. № 7. С. 38-43.
7. *Воронова Л.И.* Интеллектуальные базы данных. Учебное пособие. М.: МТУСИ, 2013. 35 с.
8. *Ковтун И.И., Романенко Г.С.* Математическое моделирование, численные методы и комплексы программ в задачах повышения эффективности многономенклатурных производств // Информатизация и связь. 2020. № 1. С. 48-54.
9. *Ковтун И.И., Романенко Г.С., Селиванов С.А.* Функционально-реляционный анализ вариантов автоматизации государственных организаций, предприятий, производств // Информатизация и связь. 2024. № 7. С. 19-26.

РЕКОМЕНДАЦИИ МСЭ-Т ПО ПРИМЕНЕНИЮ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ЭКСПЛУАТАЦИИ И УПРАВЛЕНИЯ ЭЛЕКТРОСВЯЗЬЮ

Нетес Виктор Александрович
МТУСИ, профессор, д.т.н., Москва, Россия
v.a.netes@mtuci.ru

Аннотация

Анализируются принятые в последние годы Рекомендации МСЭ-Т серии М.3xxx, посвященные применению методов искусственного интеллекта для решения задач эксплуатации и управления сетями электросвязи. В частности, рассмотрена система умной эксплуатации, управления и технического обслуживания (SOMM) и ее развитие – структура процессов эксплуатации и управления электросвязью с использованием искусственного интеллекта (АТОМ). Указаны открытые вопросы, требующие дальнейшего изучения и стандартизации.

Ключевые слова

рекомендации МСЭ-Т, искусственный интеллект, сеть управления электросвязью, функциональная архитектура, уровни интеллектуальности, оценка эффективности, обеспечение качества.

Введение

Важной тенденцией современного этапа развития информационных технологий (ИТ) является все более широкое использование методов искусственного интеллекта (ИИ). В нашей стране принята Национальная стратегия развития ИИ на период до 2030 года [1]. Внедрение технологий ИИ предусмотрено Стратегией развития отрасли связи Российской Федерации на период до 2035 года [2].

Одна из областей применения ИИ в телекоммуникациях – эксплуатация и управление сетями электросвязи. В последние годы МСЭ-Т принял целый ряд Рекомендаций, посвященных этой тематике. Они входят в серию М.3xxx «Сеть управления электросвязью» (Telecommunications Management Network, TMN). Разработка концепции TMN началась еще в середине 1980-х годов [3]. С тех пор она претерпела заметную эволюцию, отражающую как развитие самих сетей электросвязи, так возникновение новых идей и методов в ИТ, в частности применение ИИ.

В разработках МСЭ-Т по ИИ центральную роль играет структура процессов эксплуатации и управления электросвязью с использованием искусственного интеллекта (Artificial Intelligence enhanced Telecom Operation and Management, АТОМ), которая является развитием определенной ранее системы умной эксплуатации, управления и технического обслуживания (Smart Operation, Management and Maintenance, SOMM).

Данная статья посвящена рассмотрению основных положений и анализу SOMM и АТОМ, а также других аспектов применения ИИ, описанных в Рекомендациях серии М.3xxx. В частности, рассмотрены вопросы оценки уровня интеллектуальности и эффективности систем на основе АТОМ, обеспечения с их помощью качества обслуживания и др.

Умная эксплуатация, управление и техническое обслуживание (SOMM)

В 2020 году МСЭ-Т принял Рекомендацию М.3041 [4], в которой вводится система SOMM. В ней описаны ее характеристики, сценарии и функциональная архитектура. Она служит для поддержки работы служб, управления сетью и технического обслуживания инфраструктуры, как для традиционных сетей, так и для сетей, построенных на принципах SDN/NFV.

Система SOMM включает традиционные системы управления сетью и сетевыми элементами для управления не-SDN/NFV сетями, а также оркестратор/контроллер и платформу управления и оркестрации (Management and Orchestration, MANO) для сетей SDN и NFV соответственно.

Основными характеристиками SOMM являются:

- интеллектуальность,
- управление на основе данных,
- интегрированность,
- гибкость.

SOMM обладает многоуровневой функциональной архитектурой, которая является развитием традиционной логической многоуровневой архитектуры (Logical Layered Architecture, LLA) TMN,

определенной в Рекомендации М.3010 [5]. Эта архитектура SOMM включает четыре уровня (сверху вниз):

- сценариев приложений (scenario application),
- услуг управления (management service),
- конвергенции и управления данными (data convergence and management),
- управления инфраструктурой (infrastructure management).

В приложении к [4] описано соответствие между функциональной архитектурой SOMM и LLA.

На каждом уровне располагаются несколько наборов функций управления, которые соответствуют функциональным блокам операционных (управляющих) систем (Operations Systems Function, OSF). Таким образом, выделяются четыре типа OSF: SA-OSF, MS-OSF, DM-OSF, IM-OSF.

В число этих наборов входят традиционные OSF всех уровней LLA: управления бизнесом (B-OSF), услугами (S-OSF), сетью (N-OSF), сетевыми элементами (E-OSF). Но, конечно, основными являются новые наборы функций управления. В частности, для уровня сценариев приложений они соответствуют пяти типичным сценариям применения SOMM:

- умная эксплуатация;
- умное управление;
- умное техническое обслуживание;
- умный комплексный анализ;
- традиционная эксплуатация, администрирование и техническое обслуживание (Operations, Administration and Maintenance, OAM).

Эталонные (опорные) точки q и x для взаимодействия между OSF, определенные в [5], также существуют в SOMM. Точка q находится между OSF, расположенными на разных уровнях внутри системы, а x – между OSF, расположенными на одном и том же уровне разных систем SOMM.

Эксплуатация и управление электросвязью с использованием искусственного интеллекта (АІТОМ)

Через год после SOMM в 2021 году МСЭ-Т принял Рекомендацию М.3080 [6], в которой вводится общая схема процессов эксплуатации и управления электросвязью с использованием искусственного интеллекта АІТОМ, являющаяся дальнейшим развитием SOMM.

Основные предъявляемые к АІТОМ требования:

- совместимость со стандартом SOMM;
- более мощные возможности ИИ для поддержки интеллектуального управления эксплуатацией электросвязи;
- поддержка платформы, основанной на управлении данными, например, для обмена данными, интеллектуального анализа данных, корреляции данных, машинного обучения и других целей;
- поддержка интегрированной платформы, которая может применяться для существующих и будущих сетей, поддержка междоменного управления для предоставления услуг клиентам из конца в конец;
- поддержка гибкого управления, где новая операционная система может быть быстро развернута благодаря открытым сервисам и конвергентным данным, а также общей инфраструктурной платформе для операционных систем;
- поддержка использования возможностей ИИ, которые будут доступны в любой системе, как внутри, так и вне АІТОМ;
- поддержка песочницы, дающей возможность оценки конвейера ИИ;
- поддержка механизма безопасности, гарантирующего, что АІТОМ работает в безопасных условиях;
- поддержка замкнутого цикла управления деятельностью предприятия электросвязи.

Архитектура АІТОМ основана на функциональной архитектуре SOMM, к которой сверху добавлен уровень клиентоориентированного рынка, а справа – новая область – движок ИИ.

Уровень клиентоориентированного рынка предназначен для предоставления набора приложений, услуг, данных и возможностей ИИ АІТОМ внешним клиентам операторов связи, особенно корпоративным и отраслевым клиентам. Он выполняет две функции, одна из которых – предоставление стандартизированного каталога возможностей, а другая – настройка (кастомизация) возможностей.

Движок ИИ предоставляет возможности ИИ для каждого уровня функциональной архитектуры SOMM и состоит из пяти функциональных блоков, обеспечивающих:

- управление возможностями ИИ,

- оркестровка возможностей ИИ,
- обучение в песочнице ИИ,
- управление общим хранилищем моделей ИИ,
- управление вычислительной платформой движка.

Конвейер ИИ – это цепочка функциональных блоков, формируемая некоторыми функциональными узлами АИТОМ и включающая сбор данных, предварительную обработку, обучение или оркестровку модели ИИ, формулирование или применение политики и выдачу результатов. Конвейер ИИ имеет два стадии: разработки и эксплуатации.

В стадии разработки его можно использовать для обучения в песочнице и оркестровки возможностей ИИ в конкретном приложении с автономными данными, а затем в стадии эксплуатации – для работы с возможностями ИИ с онлайн-данными.

Оценка уровня интеллектуальности и эффективности

В 2023 году была принята Рекомендация М.3384 [7], посвященная оценке уровня интеллектуальности систем АИТОМ. В ней определены 6 уровней, обозначенные цифрами от 0 до 5:

- ручной (L0) – все виды рабочих процессов по эксплуатации и управлению выполняются вручную без участия системы;
- с поддержкой системы (L1) – несколько задач выполняются автоматически системой на основе АИТОМ в соответствии с заданными вручную правилами, но для решения большинства задач требуется участие человека;
- первичный интеллект (L2) – все текущие задачи автоматически выполняются системой на основе АИТОМ;
- промежуточный интеллект (L3) – в дополнение к первичному интеллекту все задачи по сбору данных автоматически выполняются системами на основе АИТОМ;
- развитой интеллект (L4) – в дополнение к промежуточному интеллекту все задачи анализа автоматически выполняются системами на основе АИТОМ;
- полный интеллект (L5) – все виды рабочих процессов по эксплуатации и управлению выполняются интеллектуально с помощью систем на основе АИТОМ.

Отнесение к тому или иному уровню осуществляется исходя степени автоматизации 5 областей (аспектов, измерений). Они образуют полный замкнутый цикл, который начинается разработкой требований и кончается их реализацией, и включает:

- отображение намерений – преобразование намерений в отношении управления и эксплуатации или бизнес-услуг в конкретные конфигурации и стратегии, а также обратная связь о фактической реализации намерений;
- сбор данных – сбор сырых данных, необходимую обработку данных и управление данными;
- анализ – анализ собранных данных для получения информации о текущей операционной среде, услугах и клиентах, что дает основу для принятия обоснованных решений;
- принятие решений – определение политики в области эксплуатации и техобслуживания в соответствии с результатами, полученными в процессе анализа, а затем разработка системы управления и эксплуатации на основе АИТОМ;
- реализация действий – ввод в эксплуатацию систем на основе АИТОМ в соответствии с политикой, определенной в процессе принятия решений.

Ранее описанный выше подход уже был введен в Рекомендации Y.3173 [8]. Там же представлено его сравнение с подобными методами, предложенными другими организациями по стандартизации: Обществом автомобильных инженеров, Ассоциацией GSM, TM Форумом, ETSI.

Объектами оценки уровня интеллектуальности в АИТОМ могут быть:

- функциональные блоки,
- наборы функций,
- система в целом.

Более детально методология оценки уровня интеллектуальности описана в Рекомендации М.3385 [9]. В частности, в ней каждая из указанных выше областей подразделяется на несколько подобластей; приводится расчетная формула, выражающая уровень интеллектуальности в виде взвешенной суммы оценок для всех подобластей; описывается подход к автоматизации процесса оценки.

В приложении к [9] представлен пример оценки уровня интеллектуальности системы обеспечения качества услуг для сетей ИМТ-2020 (5G) и последующих поколений. Для определения весовых коэффициентов при этом использован метод анализа иерархий.

В Рекомендации М.3388 [10] определяются показатели эффективности для уровня интеллектуальности АТОМ. С этой целью вводятся три области (аспекта, измерения):

- восприятие клиентов,
- предоставление услуг,
- управление сетью.

В каждой из них выделены по 4 подобласти, для каждой из которых определены показатели эффективности (от 1 до 3), всего, таким образом, получаются 17 показателей.

Метод оценки эффективности включает три этапа:

- 1) определение базовых значений показателей эффективности для каждого уровня, исходя из сложности и срочности технического совершенствования эксплуатации и управления сетью;
- 2) автоматический расчет реальных значений показателей эффективности на основе данных, генерируемых в ходе ежедневных процессов эксплуатации и управления;
- 3) окончательная оценка эффективности на основе сравнения реальных и базовых значений показателей эффективности.

Обеспечение качества обслуживания и управление восприятием клиентов

Конечная цель эксплуатации и управления сетью электросвязи – обеспечение качества обслуживания для пользователей услуг. В Рекомендации М.3390 [11] сформулированы требования к интеллектуальному комплексному анализу с использованием АТОМ, предназначенному для обеспечения качества обслуживания на уровне систем поддержки эксплуатации. Этот анализ включает:

- анализ обеспечения сетевыми ресурсами,
- анализ качества сетевых услуг,
- анализ качества обслуживания из конца в конец,
- разработку стратегии обеспечения качества обслуживания.

Кроме того, в [11] приведены требования к сбору, хранению и обработке данных, а также к репозиторию моделей обеспечения качества обслуживания. Эти модели могут быть предназначены для решения таких задач, как обнаружение аномалий, диагностика неисправностей, самовосстановление, прогнозирование неисправностей.

В приложениях к [11] приведена схема реализации всего процесса интеллектуального комплексного анализа, включающая семь этапов, а также пример его применения для обеспечения качества услуг с анализом первопричин в сети ИМТ-2020.

Рекомендации М.3389 [12] содержит требования к управлению восприятием клиентов услуг электросвязи на основе ИИ. Основные процессы такого управления включают:

- сбор данных, позволяющих оценивать и прогнозировать восприятие клиентов (их источниками может быть сама сеть, системы управления сетью, системы управления взаимоотношениями с клиентами и др.);
- хранение данных;
- обработка данных (включает в себя очистку данных, создание аннотаций и выявление особенностей);
- управление репозиторием моделей восприятия клиентов (обучение, хранение и управление различными моделями, используемыми для обеспечения и улучшения восприятия клиентов);
- мониторинг и анализ восприятия клиентов;
- разработка стратегии оптимизации восприятия клиентов.

В приложениях к [12] приведены: схема реализации всего процесса управлению восприятием клиентов, также включающая 7 этапов; 4 типичных прикладных сценария и типичный пример – оптимизация восприятия клиентов путем обеспечения качества из конца в конец.

Управление затратами на эксплуатацию сети

Для любого оператора связи чрезвычайно важны экономические аспекты его деятельности. В Рекомендации М.3386 [13] представлены требования к управлению затратами на эксплуатацию сети в рамках АТОМ. В ней дана классификация расходов на эксплуатацию сети, в основе которой выделены области управления и категорий финансовых затрат.

Примерами областей являются:

- сеть широкополосного фиксированного доступа,
- сеть радиодоступа,

- базовая сеть,
- транспортная сеть,
- инфраструктура.

В качестве категорий затрат рекомендованы:

- осмотр оборудования в местах эксплуатации,
- ремонт в местах эксплуатации,
- диагностика в местах эксплуатации,
- монтаж и приемо-сдаточные испытания,
- запасные части,
- расходные материалы,
- услуги технического обслуживания и поддержки.

Далее выделены функциональные блоки управления затратами, сформулированы требования к ним, показана их взаимосвязь со структурой АИТОМ.

В приложении к [13] приведены контейнеры ИИ по управлению затратами для стадий разработки и эксплуатации. В другом приложении приведены два примера, которые показывают применение описанной методологии для решения следующих задач:

- 1) сочетание управления затратами и устранением неисправностей для поддержания стабильности работы сети с меньшими затратами;
- 2) создание базы данных о квотах затрат путем сбора всех видов данных.

Заключение

В последние годы МСЭ-Т принял ряд Рекомендаций, описывающих применение ИИ для эксплуатации и управления сетями электросвязи. Они входят в серию М.3xxx, посвященную TMN, и в них использованы некоторые базовые положения этой концепции. Поэтому высказываемое порой мнение, что принципы TMN устарели и их знание не нужно современным специалистам, является ошибочным.

Помимо рассмотренных в предыдущих разделах есть еще Рекомендации, посвященные таким аспектам применения ИИ, как управление энергосбережением в сетях радиодоступа 5G (М.3381), обработка заказов на выполнение работ (М.3382), анализ журналов (М.3383).

К сожалению, положения всех Рекомендаций МСЭ-Т по ИИ носят в основном весьма общий и абстрактный характер. Им не хватает конкретики, необходимой для широкого и успешного практического применения. В них сформулированы многочисленные требования, но не указаны возможные пути их выполнения. Методы ИИ разнообразны, поэтому нужны рекомендации по их выбору для решения тех или иных задач. Однако на сегодняшний день они пока отсутствуют.

Физической реализацией эталонных точек являются интерфейсы (обозначаемые теми же, но главными буквами). Для обеспечения взаимодействия между системами необходима их стандартизация. Это особенно важно для интерфейса X, находящегося между системами, принадлежащими разным операторам связи. Однако спецификации интерфейсов пока отсутствуют. Хотелось бы, чтобы в дальнейшем эти пробелы были восполнены.

Вообще, некоторый «отрыв от реальности», к сожалению, характерен и для ряда других разработанных в последние годы документов, посвященных перспективам развития телекоммуникаций [14-18].

Литература

1. Национальная стратегия развития искусственного интеллекта на период до 2030 года. Утверждена указом Президента РФ от 10.10.2019 № 490 (в редакции указа Президента РФ от 15.02.2024 № 124).
2. Стратегия развития отрасли связи Российской Федерации на период до 2035 года. Утверждена распоряжением Правительства РФ от 24.11.2023 № 3339-р.
3. *Гребешков А.Ю.* Управление сетями электросвязи по стандарту TMN: Учеб. пособие. М.: Радио и связь, 2004. 155 с.
4. Recommendation ITU-T M.3041 (02/2020). Framework of smart operation, management and maintenance.
5. Recommendation ITU-T M.3010 (02/2000). Principles for a telecommunications management network.
6. Recommendation ITU-T M.3080 (02/2021). Framework of artificial intelligence enhanced telecom operation and management (АИТОМ).
7. Recommendation ITU-T M.3384 (04/2023). Intelligence levels of artificial intelligence-enhanced telecom operation and management.

8. Recommendation ITU-T Y.3173 (02/2020). Framework for evaluating intelligence levels of future networks including IMT-2020.
9. Recommendation ITU-T M.3385 (04/2023). Intelligence levels evaluation framework of artificial intelligence enhanced telecom operation and management.
10. Recommendation ITU-T M.3388 (08/2024). Effectiveness indicators for intelligence level of AI-enhanced telecom operation and management.
11. Recommendation ITU-T M.3390 (03/2025). Requirements for smart comprehensive analysis within artificial intelligence enhanced telecom operation and management (AITOM).
12. Recommendation ITU-T M.3389 (03/2025). Requirements for artificial intelligence-based customer experience management of telecom services.
13. Recommendation ITU-T M.3386 (01/2024). Requirements for the management of network operation cost within AI enhanced Telecom Operation and Management (AITOM) in telecommunication operational aspects.
14. *Нетес В.А.* Концепция МСЭ-Т «Сеть 2030»: проблемные точки // Сб. тр. XVII Междунар. отраслевой науч.-техн. конф. «Технологии информационного общества». М.: МТУСИ, 2023. С. 137-139.
15. *Росляков А.В.* Сети фиксированной связи пятого поколения. Учеб. Пособие. М.: ООО «ИКЦ «Колос-с», 2024. 232 с.
16. *Нетес В.А.* Человекоподобные сети: основные понятия и архитектура // Первая миля. 2024. № 5. С. 14-18.
17. *Нетес В.А.* О достижимости целей в развитии телекоммуникаций // Первая миля. 2025. № 3. С. 46-48.
18. *Нетес В. А.* Отказы по общей причине и их влияние на устойчивость сетей и систем связи // Т-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 2. С. 4-9. DOI 10.36724/2072-8735-2024-18-2-4-9. EDN GYGLXF.

РАСПОЗНАВАНИЕ ПЭТ-ОБЪЕКТОВ В ПОТОКЕ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

Андрюшина Марина Николаевна

МТУСИ, магистрант, группа М152501, Москва, Россия

m.n.andryushina@edu.mtuci.ru

Рожков Сергей Александрович

МТУСИ, профессор кафедры ИСУиА, д.т.н., Москва, Россия

s.a.rozhkov@mtuci.ru

Аннотация

В работе рассмотрены методы обнаружения ПЭТ-объектов в потоке: традиционные, системы с нейронными сетями, системы технического зрения. Показан подход для обнаружения ПЭТ-объектов в потоке, который моделируется сетью Хопфилда и сетью персептрона в среде Matlab.

Ключевые слова

система технического зрения, распознавание, многослойная нейронная сеть прямого действия (МНС), персептрон, сеть Хопфилда, полиэтилентерефталат (ПЭТ)

Введение

Технология сортировки твердых отходов включает в себя механические и оптические процессы для подготовки материалов к переработке. Для эффективной сортировки различных типов ПЭТ-объектов на перерабатывающих предприятиях этот процесс происходит в реальном времени. При этом задачу обнаружения твердых отходов в потоке из полиэтилентерефталата (ПЭТ) усложняет тот факт, что это объекты не только разнообразной формы, но для их обнаружения следует обеспечить специальные условия освещения [1, 2].

Основные типы ПЭТ-объектов могут быть классифицированы на стандартные бутылки объемом 0,5-2 л, которые имеют повреждения или имеют различные загрязнения. Также могут быть отклонения в форме (плоские, округлые, вытянутые), нарушения целостности (пробоины, разрывы), а также могут иметь дефекты поверхности (царапины, неровности) [3].

ПЭТ-бутылки производят из специальных хлопьев, из которых далее получают агломерат и полиэфирное волокно. Из агломерата получают композит и гранулят, а из полиэфирного волокна – синтепон, линолеум, мягкую кровлю, поддоны различного назначения, садово-парковый инвентарь и др. Композитные материалы широко используют в электротехнике, машиностроении, автомобильной технике, а из гранулята производят упаковочные ПЭТ-ленты, пленки, тару.

Методы обнаружения ПЭТ-объектов в потоке

Традиционные методы сортировки отходов, такие, как ручной отбор, простые оптические сепараторы, имеют низкую производительность и высокую долю ошибок при контроле. Современные системы сортировки включают оптические сортировщики на основе камер видимого спектра, системы с ИК-спектрометром и комплексные системы с нейронными сетями.

Известные автоматические системы [3-7] для обнаружения объектов из полиэтилентерефталата (ПЭТ) обеспечивают качество переработки за счет быстрого и точного выявления и классификации материалов. Основным инструментом таких систем являются системы технического зрения с нейронными технологиями, которые решают сложные задачи в реальном времени. Совмещение систем технического зрения и нейронных сетей позволяет создавать универсальные модели и многослойные сети для моделирования сложных зависимостей в изображениях и точной классификации ПЭТ-объектов. Нейронные сети обучаются на наборах изображений в различных условиях и точно классифицируют ПЭТ-объекты, обходя ограничения традиционных алгоритмов, работая с неполными или зашумленными данными. Преимуществом таких систем является высокая скорость и точность обнаружения ПЭТ-объектов, а также возможность адаптации для различных условий эксплуатации.

Устройства для контроля потоков ПЭТ в основном используют оптические методы, а для повышения быстродействия обнаружения используют нейронные алгоритмы.

Современные методы обнаружения ПЭТ-объектов в потоке включают в себя предварительную обработку изображения, выделение признаков с помощью детекторов ключевых точек и применение свёрточных нейронных сетей (CNN) для распознавания и классификации с учётом вариаций формы, размера и положения объектов.

Примерами применения устройства для контроля дефектов и твердых отходов в потоке являются системы сортировки объектов, где используется устройство, которое разделяет отраженный свет по длине волны на части [8, 9]. Каждая из этих частей попадает на детектор, а полученные данные обрабатываются и передаются в виде команд на устройство сортировки, работающее на сжатом воздухе.

Система также предусматривает возможность удаления объектов: в зависимости от результатов сортировки они направляются в нужные контейнеры или накопители. Объекты, удаленные из потока, могут быть направлены в общий накопитель или в различные накопители в зависимости от результатов анализа. Схема такого сканирующего устройства для контроля твердых отходов в потоке показана на рис. 1 [8].

Для идентификации материалов в условиях динамического потока устройство (4) обеспечивает спектральное разделение отраженного светового потока. Многогранное зеркало (13) вращается вокруг своей центральной оси. Источники света (5a) и (5b) состоят из двух отдельных ламп, которые расположены по обе стороны от контролируемого потока вещества (1). Источник светового излучения (6a) представляет собой линейный источник, а второй источник светового излучения (6b) – это источник точечного освещения (лазер), который может работать в импульсном или непрерывном режиме. Вращающееся многогранное зеркало (13) расположено таким образом, чтобы обеспечить сканирующее движение светового потока перпендикулярно направлению перемещению контролируемого вещества (1). Детектор (11) представляет собой спектрометр, адаптированный для сканирования всей ширины потока вещества (1).

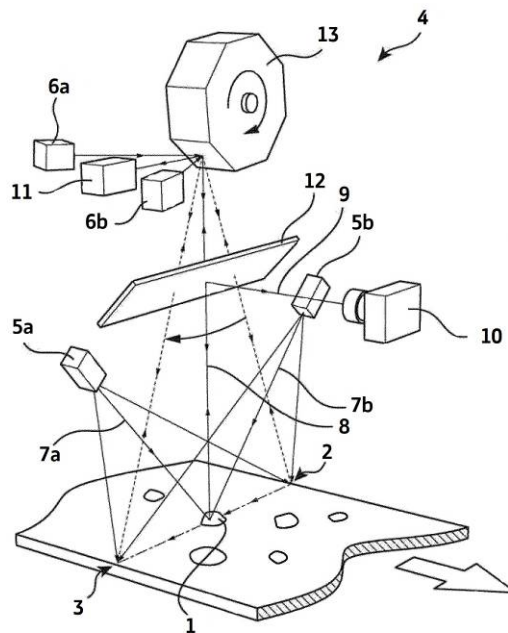


Рис. 1. Схема сканирующего устройства для контроля твердых отходов

Сложная конструкция сканирующей системы увеличивает стоимость производства и обслуживания. При этом точная настройка зеркал и сканирующих элементов требует особого внимания, так как они подвержены хроматическим aberrациям и дисперсии, что сказывается на качестве и разрешении изображения.

Относительно невысокая скорость сканирования ограничивает скорость обработки материалов в потоке, что приводит к потерям интенсивности светового потока из-за отражений и спектральных помех. Двойная оптика увеличивает размеры, вес и энергопотребление, а загрязнение и износ оптики со временем снижают контрастность и точность сканирующей системы. Механические части сканера восприимчивы к вибрациям и износу, что затрудняет поддержание равномерного освещения.

Для решения задач идентификации и управления в сканирующих системах используются многослойные нейронные сети прямого действия (МНС) или многослойные перцептроны (МП), где примером минимальной реализацией является двухслойная нейронная сеть. Дополнительное преимущество

нейронных сетей проявляется в их способности выделять общие принципы при предъявлении некоторого набора учебных векторов с неполным набором данных. Необходимо также отметить способность нейронной сети получать желаемый выход в случае неполного или нечеткого набора данных, приводящего к ошибочным результатам в случае использования традиционных компьютерных алгоритмов и программ.

Структурная схема двухслойной нейронной сети прямого действия представлена на рис. 2. В ней сигналы распространяются от входа к выходу без связей внутри слоев и обратных связей между сигналами.

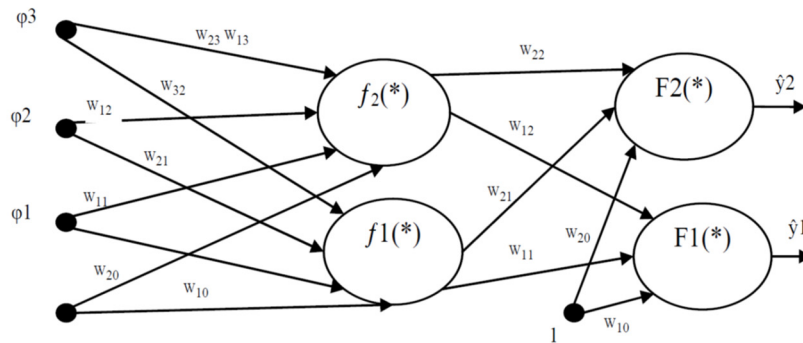


Рис. 2. Структурная схема двухслойной нейронной сети прямого действия

Математическое представление реализации модели сети следующее:

$$g_i(\theta) = y_i(\theta) = y_i(w, W) = \left(F_i \sum_{j=1}^{n_h} W_{ij} f_j \left(\sum_{l=1}^{n_\phi} w_{jl} \phi_l + w_{j0} \right) + W_{i0} \right), \quad (1)$$

где: n_ϕ – размерность вектора входов нейронной сети; n_h – число нейронов в скрытом слое; q – вектор настраиваемых параметров нейронной сети, который включает весовые коэффициенты и нейронные сдвиги (w_{jl}, W_{ij}); $f_i(x)$ – активационная функция нейронов скрытого слоя; $F_i(x)$ – активационная функция нейронов исходного слоя.

при условии соответствующего выбора активационных функций и весовых коэффициентов МНС (1) может быть использована как модельная структура для решения задачи идентификации.

В качестве систем контроля качества потоков материалов применяют системы технического зрения [4]. Системы обнаружения объектов – основная часть таких систем и предназначены для обнаружения, распознавания и классификации ПЭТ-объектов (рис. 3).

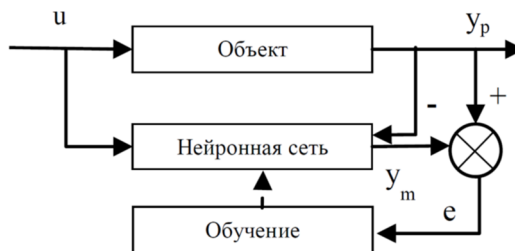


Рис. 3. Структурная схема МНС

Перед тем, как приступить к обнаружению ПЭТ-объектов в потоке с помощью нейронных сетей, необходимо преобразовать исходное изображение, полученное от системы сканирования, в набор характерных признаков. Этап извлечения признаков (feature extraction) служит критически важным звеном между «сырыми» пиксельными данными и работой классификатора. Для полутонового изображения набор характерных признаков имеет вид матрицы яркости.

Для упрощения последующего анализа целесообразно преобразовать матричное представление изображения в векторную форму путём последовательного объединения элементов матрицы в единый вектор по столбцам или по строкам. При этом всякий целевой вектор можно рассматривать как набор характеристик, означающий некоторый объект. Этот вектор содержит все элементы матрицы, последовательно считанные по столбцам [7].

Сеть Хопфилда относится к классу рекуррентных нейронных сетей, которые обладают тем свойством, что за конечное число тактов времени они из произвольного начального состояния приходят в состояние устойчивого равновесия (аттрактор). Количество таких аттракторов определяет объем ассоциативной памяти сети Хопфилда.

На вход сети предъявляется входной образ. Фактически это осуществляется непосредственной начальной (нулевой) установкой компонентов исходных сигналов:

$$y_i(0) = p_i, \quad i = \overline{1, N} \tag{2}$$

Предъявляется вектор входа и вычисляются элементы W_{ij} матрицы весов, и для активных нейронов на основании их текущего и предельного состояния вычисляется энергия каждого образа обнаружения: распознавание.

$$E = -\frac{1}{2} \sum_{i=1}^N p_i(k+1) \sum_{j=1}^N w_{ij} p_j(k) \tag{3}$$

Предъявляются образы по формулам (2), (3), вычисляются последовательные состояния и значения выходных сигналов нейронов до достижения ими устойчивых состояний.

$$z_j(k+1) = \sum_{i=1}^N w_{ij} y_i(k), \quad i = \overline{0, N} \tag{4}$$

$$y_j(k+1) = f(z_j(k+1)) = \begin{cases} 1, & \text{если } z_j(k+1) > \Theta_j, \\ 0, & \text{если } z_j(k+1) < \Theta_j, \\ y_j(k) & \text{в остальных случаях} \end{cases} \tag{5}$$

где f – предельная активационная функция; Θ_j – заданный порог.

Вызов образа из сети происходит через последовательность состояний, каждое последующее из которых имеет меньшую энергию по сравнению с предыдущим.

Этот спуск в сеть с более низким энергетическим уровнем продолжается до достижения сетью энергетического минимума, то есть до воспроизводства сетью вызываемого образа.

Алгоритм обнаружения ПЭТ-объектов в потоке. Из системы сканирования на систему обнаружения поступает изображение фрагмента потока. Для дальнейшего использования его в работе изображение подготавливается к обнаружению. Для этого изображение преобразуется в полутоновое и складывается матрица яркости.

Для облегчения анализа удобно перейти от матричного представления изображения к векторному, собирая элементы столбцов матрицы в один вектор [5], который применяется как вход искусственной нейронной сети Хопфилда. Структурная схема двухслойной нейронной сети показана на рис. 4.

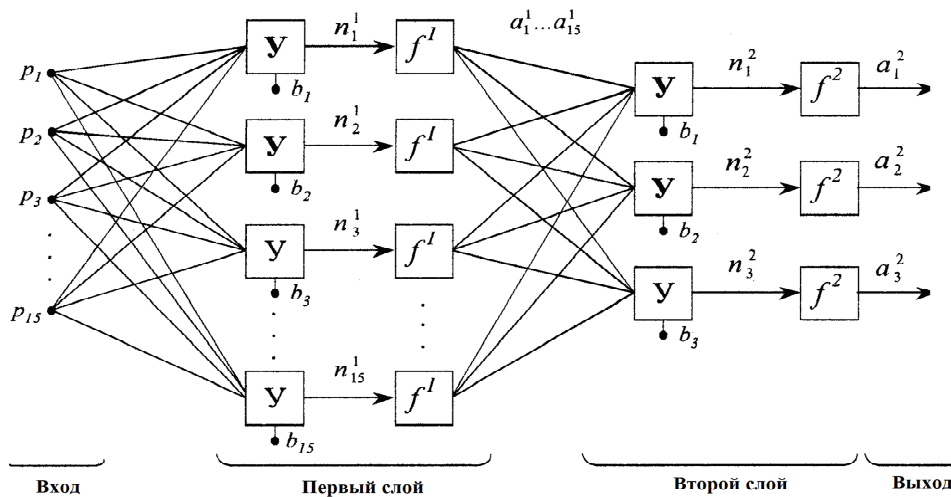


Рис. 4. Структурная схема двухслойной нейронной сети

Создается сеть Хопфилда, содержащая рекуррентный слой, в этом случае целевые векторы определяют начальное состояние первого слоя. Выход сети Хопфилда поступает на однослойный персептрон, веса и сдвиг которого необходимо настроить таким образом, чтобы решить задачу классификации входных векторов.

Для реализации этой задачи, сначала необходимо создать слой персептрона, задать значения, возможные на выходе сети. Сеть Хопфилда восстанавливает изображения, а персептрон обучается классифицировать их: на вход подается описание изображения с выхода Хопфилда и соответствующий целевой вектор. Далее персептрон настраивается так, чтобы точно идентифицировать класс каждого восстановленного изображения. Моделируется сеть персептрона и вычисляется погрешность на первом шаге настройки (обучения). Если выход соответствует поставленной цели, то сеть обучена правильно.

Если же выход не соответствует поставленной цели, то необходимо продолжать процедуру обучения сети до тех пор, пока все значения выхода не будут соответствовать поставленным целям. Эта процедура повторяется для всех возможных результатов обнаружения необходимых целей.

Когда сеть полностью обучена правильно, то можно переходить к классификации изображений ПЭТ-объектов, получаемых на разных этапах потока.

Сначала на вход сети поступает предварительно обработанное изображение, представленное в виде вектора оттенков серого. В слое Хопфилда оно обнаруживается, и на выходе этого слоя образуется вектор (точка равновесия), отвечающий конкретному ПЭТ-объекту. Далее вектор поступает на вход слоя персептрона [10].

Если персептрон был заранее обучен обнаруживать и классифицировать этот объект, то в результате на выходе сети получаем номер класса в двоичном коде, к которому принадлежит обнаруженный объект. Таким образом, процедура обнаружения закончена.

Если же на выходе сети появился номер, который не соответствует ни одному из ранее обнаруживаемых объектов, то сеть необходимо дообучить дополнительно, чтобы она правильно обнаружила и этот объект. Тогда процедура обучения повторяется заново.

После успешного обучения сеть обнаруживает и правильно классифицирует этот ПЭТ-объект. Процесс для обнаружения и классификации ПЭТ-объектов в потоке, который показал хорошее быстрое действие при реализации в среде Matlab, показан на рис. 5.

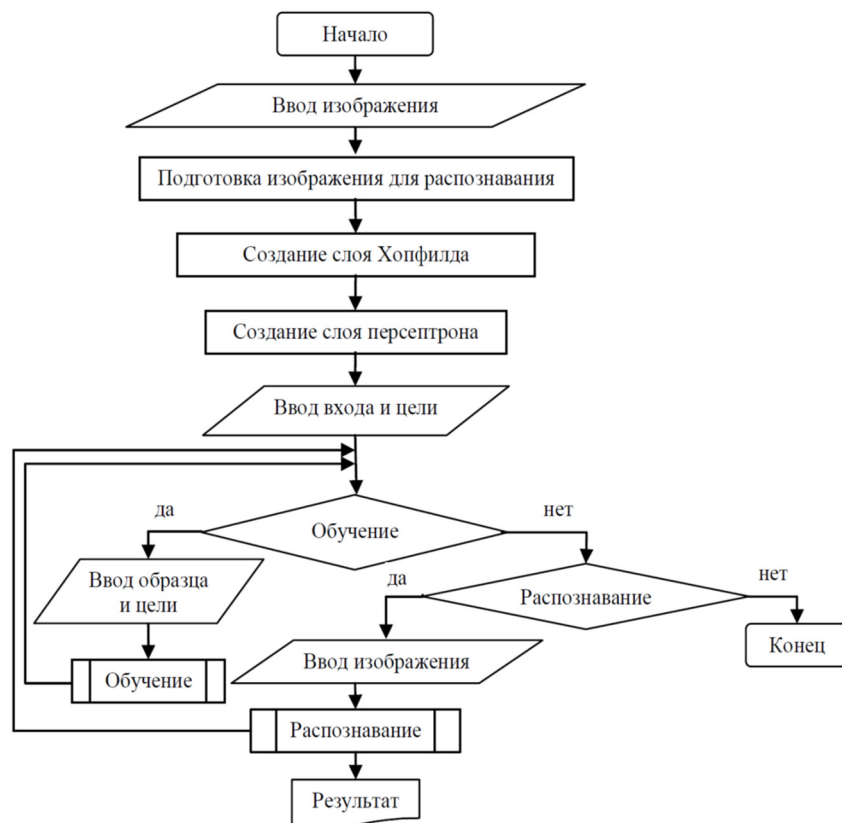


Рис. 5. Процесс для обнаружения и классификации ПЭТ-объектов в потоке

Схемы моделирования в среде Matlab показаны на рис. 6 и 7.

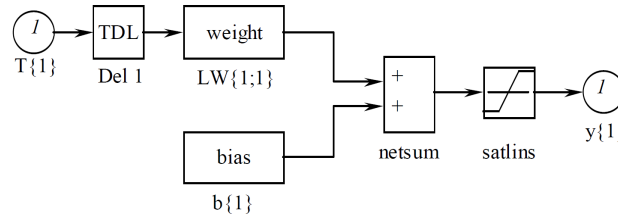


Рис. 6. Структурная схема слоя Хопфилда

Сеть Хопфилда содержит 15 нейронов, каждый из которых соединён со всеми 15 входами, образуя полностью связную архитектуру с размерностью входного вектора $n = 15$.

В данной сети применяется функция активации *satlins*, определяемая следующим соотношением [1].

$$satlins(n) = \begin{cases} -1, & \text{если } n < -1, \\ n, & \text{если } -1 \leq n \leq 1, \\ 1, & \text{если } n > 1 \end{cases} \quad (6)$$

Для потока без ПЭТ-объектов на выходе слоя Хопфилда получаем вектор, представляющий точку равновесия для данного изображения: $T = [1; -1; 1; -1; -1; 1; -1; 1; -1; 1; -1; 1; -1; -1]$.

Затем создается и моделируется сеть персептрона. Поскольку выход искусственной нейронной сети трехзначный, то слой персептрона содержит три нейрона, каждый из которых соединен с каждым нейроном на выходе слоя Хопфилда.

В слое персептрона применяется функция активации *hardlim*, определяемая следующим соотношением:

$$hardlim(n) = \begin{cases} 0, & \text{если } n < 0, \\ 1, & \text{если } n \geq 0. \end{cases} \quad (7)$$

Для того, чтобы на выходе сети были значения, соответствующие конкретным объектам, необходимо обучить персептрон правильно классифицировать обнаруживаемые изображения. Каждому типу объекта присваивается конкретный номер в двоичном коде. Поскольку производится обнаружение шести типов объектов, то достаточно трехзначного значения цели. Поток без объектов присваивается значение цели: $K = [0; 0; 0]$.

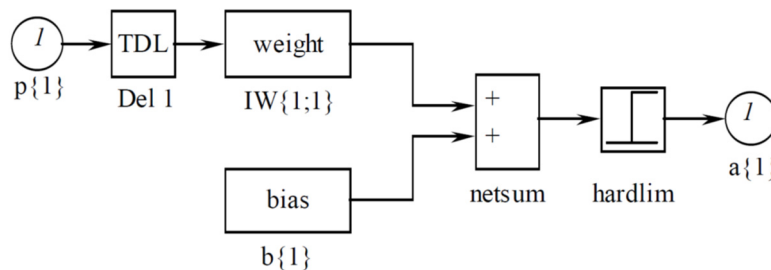









Рис. 7. Структурная схема слоя персептрона

Часть результатов моделирования обучения персептрона для классификации объектов представлены на рис.8 и в таблице 1.

Таблица 1

Результаты моделирования двухслойной сети

Вход эталон	Выход первого слоя	Выход второго слоя
	$[1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1]$	0 0 0
	$[1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1]$	0 0 1
	$[-1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1]$	0 1 1
	$[1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1]$ $1 \ 0,987 \ -0,289 \ -1 \ 1]$	1 0 0
	$[-1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1]$ $1 \ -1 \ 1 \ -1 \ 1]$	1 1 0
	$[1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1]$ $1 \ -1 \ 1 \ 1 \ -1]$	1 0 1
	$[-1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1]$ $1 \ 1 \ -1 \ -1 \ 1]$	1 1 1

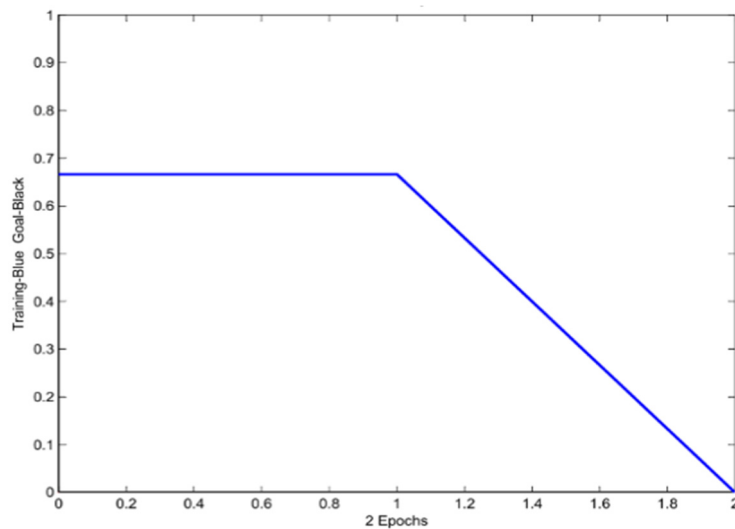


Рис. 8. График обучения персептрона

Выводы: реализация алгоритма в среде MATLAB подтвердила его высокую скорость работы при работе с неполными данными, что делает его более предпочтительным методом, чем традиционные алгоритмы в аналогичных условиях.

Заключение

Решение задачи обнаружения ПЭТ-объектов в потоке осложняется большим объемом обрабатываемых данных и разнообразием распознаваемых объектов. Применение нейронных сетей (НС) позволяет снизить размерность задачи распознавания, сохраняя свойства делимости классов и обеспечивая высокую скорость обработки. Это обеспечивает эффективное выявление ПЭТ-объектов в реальном времени в потоке. В ходе практики разработан алгоритм распознавания на основе двухслойной нейронной сети, включающей элементы сети Хопфилда и персептрона. Алгоритм, разработанный на основе сети Хопфилда, обеспечивает программирование НС без этапа обучения. Он реализован и протестирован в программной среде MATLAB. Полученные от него результаты подтверждают перспективность выбранного подхода и готовность к дальнейшей разработке системы.

Литература

1. *Kou W.* Possibilities of utilizing polyethylene terephthalate (PET) bottle deposit-refund system in China», Ph.D. dissertation, Dept. Environ. Technol., Lappeenranta-Lahti Univ., Lappeenranta, Finland, 2019.
2. *Tan Z. et al.* Identification for Recycling PET Plastic Bottles by Polarization Vision // IEEE Access, Vol.9. 11 January 2021. pp. 27510-27517. DOI: 10.1109/ACCESS.2021.3050816 <https://ieeexplore.ieee.org/document/9319667>.
3. *Писаревский А.Н., Чернявский А.Ф., Афанасьев Г.К.* и др. Системы технического зрения (принципиальные основы, аппаратное и математическое обеспечение); Под общ. ред. А.Н. Писаревского, А.Ф. Чернявского. Л.: Машиностроение. Ленингр. отд-ние, 1988. 424 с.
4. *Катыс Г.П.* Восприятие и анализ оптической информации автоматической системой. М.: Машиностроение, 1986. 416 с.
5. Методы робастного, нейро-нечеткого и адаптивного управления: Учебник / Под ред. Н.Д. Егупова; изд. 2-ое, стереотипное. М: Изд-во МГТУ им. Н.Э. Баумана, 2002. 744 с.
6. *Прэнтт У.* Цифровая обработка изображений: Пер. с англ. М.: Мир, 1982. Кн. 1. 312 с.
7. *Рожков С.А., Бражник Д.А.* Использование нейросетевых структур для построения систем распознавания образов // Автоматика. Автоматизация. Электротехнические комплексы и системы. 2004. № 2(14). С. 247-253.
8. WO 2015/063300 A1. Inspection apparatus: заявка PCT/EP2014/073578 / заявитель TOMRA Sorting AS; опубл. 07.05.2015. Режим доступа: <https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2015063300.freepatent>
9. А.с. 1839510 СССР, МКИ D06H3/08 С.А. Рожков, К.В. Тимофеев, А.П. Храпливый, А.М. Бражник (СССР). Устройство для обнаружения дефектов движущегося полотна ткани с печатным рисунком. №4771927/12; Заявл. 19.12.89; д.с.п.
10. *Медведев В.С. Потемкин В.Г.* Нейронные сети. MATLAB 6 / Под общ. ред. к.т.н. В.Г. Потемкина. М.: ДИАЛОГ-МИФИ, 2002. 496 с.

ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ РАДИОРЕСУРСОВ И ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ ДЛЯ РАЗНОРОДНОГО ОБРАЗОВАТЕЛЬНОГО ТРАФИКА В ПЕРСПЕКТИВНЫХ БЕСПРОВОДНЫХ СЕТЯХ

Кархили Ханин

МТУСИ, студент, Москва, Россия,
haninkarhili@yandex.com

Давуд Таммам

МТУСИ, ассистент кафедры ССuСК, Москва, Россия,
t.davud@mtuci.ru

Степанов Сергей Николаевич

МТУСИ, заведующий кафедрой ССuСК, д.т.н., Москва, Россия,
s.n.stepanov@mtuci.ru

Аннотация

В условиях цифровизации образования беспроводные сети становятся ключевой инфраструктурой, обеспечивающей доступ к учебным ресурсам, интерактивным занятиям и интеллектуальным сервисам. Современные образовательные приложения формируют разнородный трафик с существенно различающимися требованиями к пропускной способности, задержке, надежности и вычислительным ресурсам. В статье рассматривается функциональная модель распределения радиоресурсов и вычислительных ресурсов в узле беспроводного доступа, ориентированная на обслуживание образовательных сервисов. Описывается механизм поступления и обслуживания заявок, принципы приоритизации и алгоритм распределения ресурсов, обеспечивающий устойчивое качество образовательного процесса.

Ключевые слова

качество обслуживания (QoS), гетерогенные ресурсы, совместное планирование, радиоресурсы, вычислительные ресурсы, гетерогенный образовательный трафик, приоритизация.

Введение

В контексте цифровой трансформации образовательных систем новые технологии следует рассматривать как результат конвергенции телекоммуникационных сетей, вычислительных платформ и интеллектуальных сервисов, формирующих распределённую и высокодоступную учебную среду.

Развитие широкополосных и мобильных сетей связи, включая архитектуры 5G и перспективные решения с низкой задержкой, обеспечивает устойчивую передачу мультимедийного контента, поддержку удалённых лабораторий и синхронное взаимодействие в реальном времени. В этих условиях технологии дополненной и виртуальной реальности (AR/VR), а также голографические системы визуализации становятся практическими инструментами иммерсивного обучения, позволяя воспроизводить сложные технические процессы, пространственные структуры и экспериментальные сценарии без физического присутствия. На уровне инфраструктуры облачные платформы и алгоритмы искусственного интеллекта обеспечивают масштабируемую обработку данных, персонализацию образовательных траекторий и адаптивное управление учебными потоками [1, 2].

Основная проблема — неоднородность требований образовательных сервисов к сетевым и вычислительным ресурсам. Применение универсальных подходов к распределению ресурсов приводит к тому, что чувствительные интерактивные приложения теряют качество из-за конкуренции с менее критичным трафиком. Следовательно, требуется функциональная модель узла доступа, которая учитывает специфику образовательных приложений и обеспечивает согласованное распределение доступных ресурсов в зависимости от класса сервиса [3, 4].

Критически важно предотвратить ситуации, когда запрос приоритизированного сервиса (например, VR-тренажёра) получает радиоресурс, но затем блокируется из-за отсутствия доступных ресурсов на MEC-сервере (Multi-access Edge Computing), или наоборот. В отличие от классических моделей, ориентированных на один тип ресурсов, предлагаемая модель реализует совместное распределение ресурсов. Это позволяет проактивно моделировать и оценивать нагрузку на оба пула — радиодоступ и вычисления — и тем самым минимизировать вероятность комплексного отказа сервисов.

Разнообразие требований образовательных приложений

Современная цифровая трансформация образования приводит к появлению широкого спектра приложений, каждое из которых предъявляет уникальные и зачастую противоречивые требования к инфраструктуре связи. Эта неоднородность особенно очевидно проявляется в контексте перспективных беспроводных сетей, где необходимо одновременно обеспечивать и крайне низкие задержки, и высочайшую пропускную способность, и массовое подключение устройств. Ключевая сложность для оператора заключается в необходимости динамического и оптимального распределения ограниченных радиоресурсов и вычислительных ресурсов на границе сети (МЕС) между этими разнородными сервисами. Например, в рамках одной образовательной экосистемы могут одновременно функционировать фоновый анализ данных с тысяч датчиков, требующий высокой энергоэффективности и надежности, и интерактивный голографический класс, для которого критичны пиковая полоса пропускания и задержка на уровне миллисекунд. Такое пересечение приложений делает невозможным применение унифицированных политик управления ресурсами и требует интеллектуальных механизмов, способных классифицировать трафик, прогнозировать его поведение и выделять соответствующие ресурсы в режиме реального времени [5-13].

Детальное рассмотрение услуг, описанных в рекомендации МСЭ-Т Y.3117, позволяет четко структурировать их требования [5]:

1) **Интерактивный класс:** базовый сервис, основанный на двусторонней передаче видео и аудио между преподавателем и учениками. Его основные требования – стабильная полоса пропускания для каждого видеотерминала (≥ 8 Мбит/с) и умеренная задержка ($\leq 150-200$ мс) для обеспечения плавного диалога. Потребность в МЕС минимальна и может ограничиваться кэшированием контента.

2) **VR/AR класс:** иммерсивный сервис, создающий виртуальную образовательную среду. Его отличительная черта – экспоненциальный рост требований к пропускной способности (от 80 Мбит/с до 1 Гбит/с на терминал) и жесткие ограничения по задержке ($\leq 8-20$ мс) в зависимости от целевого уровня качества восприятия (удовлетворительное, комфортное, идеальное). Для рендеринга сложной 3D-графики и минимизации сетевой задержки критически необходимы распределенные вычислительные ресурсы МЕС.

3) **Голографический класс:** наиболее требовательный сервис, представляющий собой эволюцию VR/AR в сторону полного погружения с использованием светового поля и 6DoF (6 степеней свободы). Для создания эффекта присутствия требуются экстремальные значения пропускной способности (до 10 Гбит/с) и ультранизкой задержки (≤ 1 мс). Обработка голографических данных невозможна без мощных МЕС-узлов, расположенных в непосредственной близости от пользователей.

4) **Умный кампус (HD-видеонаблюдение):** сервис обеспечения безопасности, характеризующийся большим количеством распределенных камер, передающих видеопотоки высокого разрешения. Требуется гарантированной полосы пропускания (≥ 8 Мбит/с на камеру) и умеренной задержки (≤ 150 мс). МЕС здесь используется для интеллектуального анализа видео (распознавание лиц, инцидентов) на границе сети, что снижает нагрузку на центр обработки данных.

5) **Анализ поведения (датчики, браслеты):** сервис сбора телеметрии, ориентированный на надежность и своевременность доставки небольших объемов данных с множества IoT-устройств. Ключевые требования включают низкую задержку (≤ 30 мс) и высокую надежность (потеря пакетов $\leq 0.01\%$). МЕС выполняет важную функцию предварительной обработки, агрегации и фильтрации данных перед их отправкой в облако для углубленного анализа.

6) **Управление образованием (платформы):** массовый сервис широкополосного доступа к образовательным ресурсам и платформам совместной работы. Основная нагрузка ложится на ядро сети, требующее обеспечения высокой пропускной способности и поддержки фиксированно-мобильной конвергенции (FMC). Необходимость в МЕС-ресурсах для этого сервиса относительно невысока.

Сводные требования основных классов образовательных приложений к сетевым ресурсам могут быть представлены в виде таблицы (табл. 1).

Таким образом, разнородность образовательного трафика, охватывающая диапазон от фоновых IoT-данных до интерактивного голографического видео, обуславливает необходимость разработки гибкой и интеллектуальной функциональной модели. Эта модель должна динамически распределять как радиоресурсы, так и вычислительные ресурсы МЕС-узлов в зависимости от конкретных требований SLA (Service Level Agreement – соглашение об уровне обслуживания) каждого типа образовательного сценария.

Описание функциональной модели узла доступа

Функциональная модель узла доступа в современных (умных) образовательных учреждениях рассматривается как система обслуживания с конечным числом источников заявок, соответствующих активным пользовательским терминалам и устройствам образовательной среды. Каждый источник в произвольные моменты времени формирует заявку на обслуживание, связанную с иницированием или продолжением конкретного образовательного сервиса определённого класса. Заявки классифицируются по типам трафика и отличаются требуемым объёмом радиоресурсов, а также потребностью в вычислительных ресурсах МЕС (процессоры CPU, графические процессоры GPU, память).

Таблица 1

Требования к радиоресурсам и МЕС для различных сервисов умного образования [5, 6]

Приложение	Тип сервиса	Количество терминалов	Требуемая пропускная способность на терминал	Допустимая задержка (Latency)	Потребность в МЕС
Интерактивный класс (видео, аудио)	Видеотерминал	3-4 на класс	≥ 8 Мбит/с	≤ 150 мс	Низкая (возможно кэширование)
	Преподавательский терминал	1 на класс	≥ 2 Мбит/с	≤ 200 мс	
VR/AR класс	Удовлетворительный опыт	30-50 на класс	80 Мбит/с	≤ 20 мс	Высокая (для рендеринга и снижения задержки)
	Комфортный опыт	30-50 на класс	260 Мбит/с	≤ 15 мс	
	Идеальный опыт	30-50 на класс	1 Гбит/с	≤ 8 мс	
Голографический класс	Иммерсивный опыт	30-50 на класс	100 Мбит/с	≤ 10 мс	Критическая (обработка светового поля, синхронизация)
	Интерактивный опыт	30-50 на класс	1 Гбит/с	≤ 5 мс	
	Идеальный опыт	30-50 на класс	10 Гбит/с	≤ 1 мс	
Умный кампус	HD-видеонаблюдение	Множество, распределенные	≥ 8 Мбит/с	≤ 150 мс	Средняя (анализ видео на границе)
Анализ поведения (анализ интеллектуального образования)	Датчики	6 на площадке	≥ 2 Мбит/с	≤ 30 мс	Высокая (предобработка и агрегация данных)
	Умный браслет	30-50 на класс	≥ 256 Кбит/с	≤ 30 мс	
Управление образованием	Приложения (для обучения, преподавания, работы в офисе)	Массовый одновременный доступ	≥ 25 Мбит/с	≤ 30 мс	Низкая (централизованная обработка)

Каждый источник в зависимости от образовательного сценария может находиться в активном или неактивном состоянии и при переходе в активное состояние инициирует заявку на обслуживание соответствующего класса.

Поток попыток установления связи носит состояние-зависимый характер: интенсивность появления новых заявок уменьшается по мере роста числа уже активных сессий, поскольку каждый активный источник временно «выбывает» из пула потенциальных инициаторов.

Процесс генерации заявок носит стохастический характер, однако его интенсивность ограничена числом свободных источников, что отражает реальное поведение пользователей в образовательной среде и исключает неограниченный рост нагрузки. Узел доступа реализует механизм совместного допуска, при котором каждая заявка проверяется одновременно по двум ресурсным ограничениям: наличию требуемого объёма радиоресурсов в обслуживающей соте и доступности соответствующих вычислительных ресурсов на МЕС. Обслуживание заявки начинается только при условии одновременного резервирования обоих типов ресурсов; в противном случае заявка отклоняется без постановки в очередь, поскольку для большинства приоритетных образовательных сервисов (VR/AR-классы, голографические системы, интерактивные занятия) ожидание приводит к недопустимому ухудшению качества восприятия и нарушению SLA. В течение всего времени сеанса обслуживания заявка удерживает выделенные ресурсы, после завершения которого они немедленно освобождаются и возвращаются в общий пул.

В нашем исследовании мы предполагаем, что узел доступа представляет собой соту связи в пределах небольшого университетского кампуса. На территории этого кампуса находится лаборатория виртуальной реальности с ограниченным числом студентов (от 30 до 50) и мультимедийные интерактивные учебные аудитории (с трансляцией видео в разрешении 4К). В кампусе установлены камеры видеонаблюдения высокого разрешения и датчики для измерения параметров, таких как температура, влажность, качество воздуха и другие. Для мониторинга результатов образовательного процесса в виртуальной лаборатории каждый студент носит умный браслет, измеряющий его биометрические показатели. Информация, собираемая с помощью этих браслетов, позволяет определить степень вовлеченности и понимания материала каждым студентом, а также адаптировать учебный процесс и контент в соответствии с когнитивными способностями каждого учащегося.

Следовательно, заявки могут быть классифицированы на три класса обслуживания: сеансы VR (высший приоритет), 4К-лекции/видео (средний приоритет) и эластичные данные от различных типов датчиков (низший приоритет). Заявки AR/VR и 4К-видео для своего обслуживания требуют выделения как физических радиоресурсов (PRB), так и вычислительных ресурсов на границе сети (MEC); для эластичных данных требуются только радиоресурсы (PRB).

В лаборатории виртуальной реальности (VR) имеется ограниченное количество устройств (например, 40 устройств), поэтому количество источников заявок для VR класса также ограничено этим количеством устройств N_1 .

Поскольку количество студентов в университете ограничено и известно, то количество одновременно активных пользователей, работающих с видео 4К в час пик, также будет ограничено долей от общего числа студентов N_2 (например, можно предположить, что возможное количество источников заявок для видео 4К составляет $N_2 = 5\%$ от общего числа студентов).

Кроме того, поскольку территория университета ограничена (следовательно, количество датчиков также ограничено) и количество умных браслетов ограничено, то количество источников заявок в третьем классе N_3 (класс эластичных данных IoT) является ограниченным.

Данная система может быть смоделирована как непрерывная цепь Маркова, где состояние представляет собой вектор активных сеансов по классам $n = (n_1, n_2, n_3)$, где классы 1, 2 и 3 соответствуют VR, 4К-видео и эластичным данным соответственно, а n_1, n_2, n_3 – количество активных сеансов в классах 1, 2 и 3 соответственно (где: $n_1 \leq N_1, n_2 \leq N_2, n_3 \leq N_3$).

Таким образом, заявка, принадлежащая к классу i (где $i = 1, 2, 3$), характеризуется вектором требуемых ресурсов $b_i = (r_i, c_i)$, где r_i – требуемый объем радиоресурсов, а c_i – требуемый объем вычислительных ресурсов.

Поведение каждого узла в сети, выступающего в роли источника трафика класса i , описывается с помощью двух состояний:

- **активное состояние (передача):** узел находится в этом состоянии, когда передает данные и занимает ресурсы. Время, которое узел проводит в этом состоянии подчиняется экспоненциальному распределению с параметром μ_i . Таким образом, среднее время обслуживания заявки равно $h_i = \frac{1}{\mu_i}$.

- **состояние ожидания (idle):** в это состояние клиент попадает после завершения обслуживания или после неудачной попытки, когда он получает отказ (блокировку). Здесь он остается до следующей попытки. Предполагается, что длительность этого состояния подчиняется экспоненциальному распределению с параметром λ_i .

Обозначим общее количество радиоресурсов в узле доступа как R , а общее количество вычислительных ресурсов в MEC как C .

Функциональная модель узла доступа описана на рисунке 1.

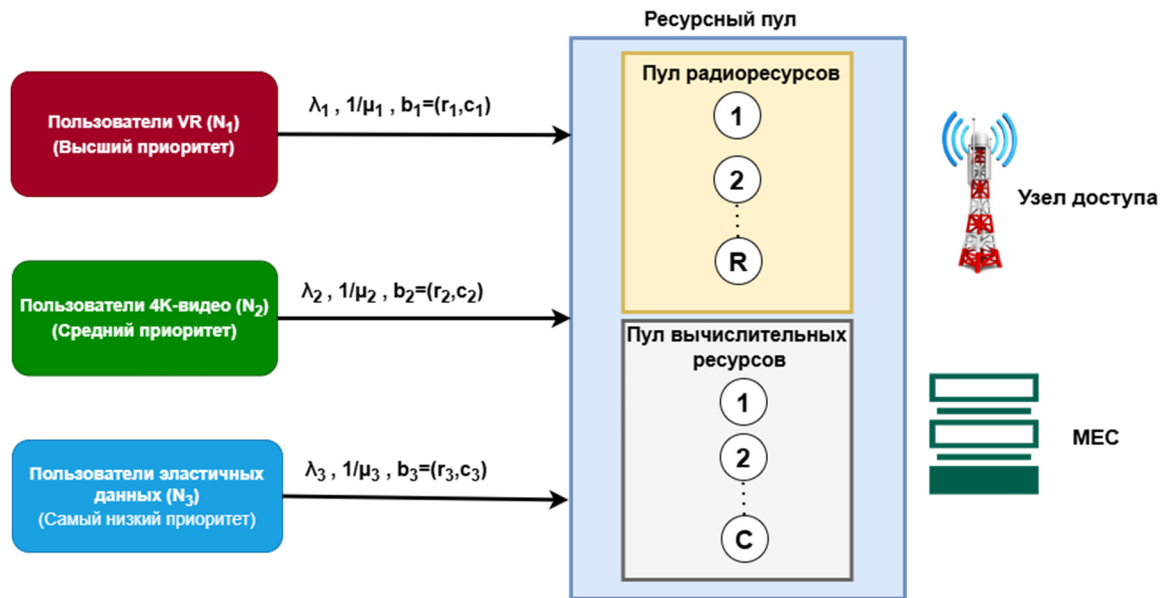


Рис. 1. Функциональная модель совместного распределения ресурсов для разнородного образовательного трафика

Таким образом, узел доступа функционирует как система с потерями, с конечным числом источников и жёстко ограниченными ресурсами, причём ограничение носит многомерный характер и определяется одновременно радиочастотным спектром и вычислительной мощностью MEC. Указанные свойства – конечное число источников, зависимость интенсивности поступления заявок от числа активных сеансов и отсутствие очередей – позволяют адекватно описывать данную функциональную модель в рамках модели Энгсета, что создаёт основу для аналитической оценки вероятностей блокирования и эффективности совместного распределения радиоресурсов и вычислительных ресурсов в образовательных беспроводных сетях.

Алгоритм работы (алгоритм распределения ресурсов)

Алгоритм работы системы состоит из следующих этапов:

- 1) классификация и приоритизация заявок;
- 2) одновременная проверка доступности ресурсов (совместный допуск). Для каждой заявки одновременно проверяются два пула: наличие PRB в обслуживаемой соте и доступный объём MEC-ресурсов. Обслуживание начинается только при условии одновременного резервирования требуемых объёмов в обоих пулах; в противном случае заявка отклоняется (без постановки в очередь);
- 3) резервирование и удержание ресурсов. После положительного решения ресурсы физически резервируются на время сессии; в течение сессии заявка удерживает выделенные PRB и MEC, после завершения – ресурсы немедленно освобождаются и возвращаются в общий пул;
- 4) в условиях дефицита возможны две опции: (а) отказ подаваемой заявки (по умолчанию для приоритетных, чувствительных к задержке сервисов – без очереди), (б) контролируемая деградация качества для сервисов с пониженным приоритетом (уменьшение полосы, перевод некоторых вычислительных задач в облако), если это не нарушает SLA критичных сервисов. Решение принимается на основе текущего профиля нагрузки и заранее заданных политик приоритетов;
- 5) для оценки вероятностей блокирования и оптимизации параметров алгоритма применяется модель с конечным числом источников (модель Энгсета) и марковская формализация состояний системы.

Заключение

В этой статье предложена функциональная модель узла беспроводного доступа, рассчитанная на совместное и согласованное распределение радиоресурсов и вычислительных ресурсов MEC для обслуживания разнородного образовательного трафика. Модель учитывает класс-ориентированную природу услуг (от фоновых IoT-поток до иммерсивных VR сессий), конечное число источников трафика и необходимость одновременного резервирования двух типов ресурсов, что минимизирует риск при

выделении только одного из необходимых ресурсов.

Ключевое практическое решение — механизм совместного допуска, при котором обслуживание начинается исключительно при одновременном подтверждении доступности PRB и вычислительных мощностей MEC. Такой подход обеспечивает соблюдение SLA для чувствительных к задержке и пропускной способности сервисов и снижает вероятность комплексных отказов, возникающих из-за несогласованного выделения ресурсов. Алгоритм распределения включает классификацию заявок по приоритетам, проверку двух пулов ресурсов, резервирование на время сессии и управляемые опции в условиях дефицита (отказ или контролируемая деградация качества).

Результаты могут служить основой для проектирования узлов доступа нового поколения, ориентированных на работу с интенсивным и разнородным образовательным трафиком.

Литература

1. Степанов М. С., Давуд Т., Степанов С. Н. Преодоление барьеров внедрения AR/VR в образовании // Методические вопросы преподавания инфокоммуникаций в высшей школе. 2025. Т. 14, № 4. С. 45-50. EDN UJAWHJ.
2. Давуд Т., Степанов М. С., Степанов С. Н. Сравнительный анализ технологий дополненной и виртуальной реальности (AR/VR) при использовании в современных образовательных процессах // Методические вопросы преподавания инфокоммуникаций в высшей школе. 2025. Т. 14, № 4. С. 51-56. EDN SXQPFJ.
3. Dawood T. Z., Muzata A. R., Stepanov M. S., Singh G., Kanishcheva M. G. The Impact of Heterogeneity on Network Resource Allocation Process for Managing Education in Real Time // 2025 Systems of Signals Generating and Processing in the Field of on Board Communications. 2025. P. 1-9. DOI: 10.1109/IEEECONF64229.2025.10948040.
4. Dawood T. Z., Stepanov S. N., Stepanov M. S., Kanishcheva M. G., Shishkin M. O. The analysis of heterogeneous traffic servicing in the cellular internet of things access node // 2024 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO). 2024. P. 1-10. DOI: 10.1109/SYNCHROINFO61835.2024.10617541.
5. International Telecommunication Union (ITU). Recommendation ITU-T Y.3117 (05/2023) – Quality of Service Assurance-Related Requirements and Framework for Smart Education Supported by IMT-2020 and Beyond [Электронный ресурс]. 2023. URL: <https://www.itu.int/epublications/es/publication/itu-t-y-3117-2023-05-quality-of-service-assurance-related-requirements-and-framework-for-smart-education-supported-by-imt-2020-and-beyond> (дата обращения: 17.01.2026).
6. Huawei Technologies Co., Ltd. Smart education scenarios white paper [Электронный ресурс]. 2018. URL: https://www-file.huawei.com/-/media/corporate/pdf/ilab/2018/smart_education_scenarios_white_paper_en_v1.pdf (дата обращения: 17.01.2026).
7. Anrabi U. M., Kanishcheva M. G., Stepanov S. N. Observation system resource planning in presence of access control based on volume of resource occupied by traffic flows // T-Comm: Телекоммуникации и транспорт. 2022. Vol. 16, No. 8, pp. 54-62. DOI 10.36724/2072-8735-2022-16-8-54-62. EDN LEVRGH.
8. Маслов А. А., Себекин Г. В., Степанов М. С. и др. Моделирование процессов обслуживания трафика реального времени в мультисервисных широкополосных сетях спутниковой связи на базе космических аппаратов на низких и средних круговых орбитах // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 12. С. 4-15. DOI 10.36724/2072-8735-2025-19-12-4-15. EDN DVDQIJ.
9. Николайчук Н. В., Степанов С. Н. Обзор принципов обработки заявок в центрах обработки данных // Научно-технические технологии в космических исследованиях Земли. 2025. Т. 17, № 6. С. 4-18. DOI 10.36724/2409-5419-2025-17-6-4-18. EDN EDEXHI.
10. Maslov A. A., Sebekin G. V., Stepanov S. N. et al. Model of processes for joint maintenance of real-time multi-service traffic and elastic data traffic in a network of low-power mobile subscriber terminals based on high-throughput satellites // T-Comm: Телекоммуникации и транспорт. 2024. Vol. 18, No. 3, pp. 41-49. DOI 10.36724/2072-8735-2024-18-3-41-49. EDN UFIBHG.
11. Лялин А. О., Степанов С. Н., Степанов М. С. Модели и методы снижения потерь заявок в мультисервисных системах связи // Системы синхронизации, формирования и обработки сигналов. 2025. Т. 16, № 1. С. 14-19. EDN GPMCWE.
12. Степанов С. Н., Слатина О. А., Трибрат Д. О. Построение и анализ модели выгрузки трафика // Телекоммуникации и информационные технологии. 2024. Т. 11, № 1. С. 13-18. EDN YATJTY.
13. Stepanov S. N., Ndayikunda Ju., Kanishcheva M. G. Resource allocation model for LTE technology with functionality of NB-IoT and reservation // T-Comm: Телекоммуникации и транспорт. 2021. Vol. 15, No. 11, pp. 69-76. DOI 10.36724/2072-8735-2021-15-11-69-76. EDN XJOAUZ.

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОГНОЗИРОВАНИЯ ДЕФЕКТОВ В ИСХОДНОМ КОДЕ С ПОМОЩЬЮ МОДИФИЦИРОВАННОГО АЛГОРИТМА XGBOOST

Фатхулин Тимур Джалилевич

*Московский технический университет связи и информатики,
доцент кафедры ИАД, к.т.н., Москва, Россия*
t.d.fatkhulin@mtuci.ru

Мяличева Альбина Андреевна

*Московский технический университет связи и информатики,
студентка группы МБД2332, Москва, Россия*

Сахарова Анастасия Михайловна

*Московский технический университет связи и информатики,
студентка группы БФИ2202, Москва, Россия*

Аннотация

Данная работа представляет анализ вариантов модификаций алгоритма машинного обучения XGBoost. Целью работы является повышение эффективности прогнозирования дефектов в исходном коде при применении модифицированной версии алгоритма XGBoost. В ходе исследования проведены эксперименты по обучению моделей XGBoost с применением рассмотренных модификаций. Проведен анализ результатов экспериментов и сделан вывод о наиболее подходящем способе модификации алгоритма для решения задачи прогнозирования дефектов в исходном коде.

Ключевые слова

прогнозирование, дефект, исходный код, модель машинного обучения, оптимизация гиперпараметров, кросс-валидация.

Введение

Алгоритм XGBoost – это оптимизированная версия алгоритма градиентного бустинга, основанная на ансамблях деревьев решений, предназначенная для задач классификации или регрессии [4-17]. В области прогнозирования дефектов в исходном коде ПО, а именно при решении задачи классификации кода на классы: «содержащий ошибки» или «без ошибок» – данный алгоритм показывает наилучший прогноз, а также показатели скорости и сложности настройки гиперпараметров по сравнению с другими традиционными методами машинного обучения [11]. Однако данный алгоритм имеет и недостатки, среди которых склонность к переобучению, неэффективная структура ансамбля, неподходящее сочетание значений гиперпараметров [11, 17]. Кроме того, значительную роль в обучении играет качество обучающих и тестовых данных [11, 13-23].

В данном исследовании рассматриваются и апробируются способы модификации алгоритма XGBoost для решения задачи классификации программного кода.

Варианты модификаций алгоритма XGBoost

В ходе исследования было отобрано несколько стратегий улучшения XGBoost.

1. Настройка гиперпараметров обучения XGBoost

Среди ключевых параметров обучения модели, влияющих на ее обобщающую способность и исключающих возможность переобучения, можно выделить [1]:

- `learning_rate` – значение, определяющее шаг обновления весов и уменьшающее прогноз каждого дерева, вносимого в ансамбль.
- `n_estimators` – количество отдельных деревьев, которые объединяются для формирования модели.
- `max_depth` – максимальная глубина дерева в составе ансамбля.
- `subsample` – доля обучающих данных, выбранных для каждого дерева в ансамбле.

- `min_child_weight` – минимальная сумма весов экземпляров, необходимая для образования дочернего узла при разделении дерева.
- `alpha` – гиперпараметр, отвечающий за силу коэффициента L1-регуляризации, применяемого к весам модели.
- `lambda` – гиперпараметр, отвечающий за силу коэффициента L2-регуляризации, применяемого к весам модели.

2. Определение важности признаков датасета

Определение важности признаков позволяет отобрать в большей степени значимые признаки, влияющие на целевую переменную при прогнозировании. Так как модель на основе XGBoost показывает высокие результаты по сравнению с моделями, обученными с применением других алгоритмов, то можно сделать вывод, что XGBoost отличается устойчивостью к вкладу неважных признаков в прогноз модели. Однако игнорирование данного влияния может снизить эффективность модели.

Встроенные методы XGBoost для определения важности признаков являются подходящим решением, потому что они наиболее совместимы с применяемым алгоритмом и не требуют использования дополнительных инструментов, а также датасеты исследуемой предметной области отличаются сравнительно небольшим числом признаков, поэтому применение встроенных методов является достаточным для решения рассматриваемой прикладной задачи.

В реализации алгоритма XGBoost имеются встроенные показатели важности признаков, которые рассчитываются во время обучения модели. К их числу относятся [1]:

- `weight` – число раз, когда признак использовался для разделения деревьев на узлы.
- `gain` – средний прирост по метрике оценки модели, который дает признак при участии в разделении дерева.
- `cover` – среднее количество экземпляров, которые проходят через разделение на узлы с указанным признаком.
- `total_gain` – общий прирост по метрике оценки модели, который дает признак при участии в разделении дерева.
- `total_cover` – общее количество экземпляров, которые проходят через разделение на узлы с указанным признаком.

3. Кросс-валидация

Кросс-валидация, или перекрестная проверка, применяется для борьбы с переобучением и повышением обобщающей способности модели. Обучение происходит N раз, в течение которых модель тестируется на разных блоках данных, количество которых равно N . В итоге все результаты прогноза модели усредняются, давая конечную оценку качеству модели [2].

Таким образом, для модификации применяемого алгоритма машинного обучения было предложено несколько вариантов, однако стоит учитывать, что рассмотренные стратегии улучшения алгоритма могут как повысить эффективность модели, так и ухудшить ее. В связи с этим необходимо предварительно провести обучение модели без применения модификаций и полученные результаты сравнить с результатами моделей, обученных с применением данных модификаций. Качество прогноза и эффективность моделей будет оцениваться и сравниваться по следующим метрикам оценки: **точность (precision), полнота (recall), F1-мера (F1 score)** [1-5].

Описание набора данных

За основу был взят архив датасетов проектов на Java [3], объединенных в один. Сборный датасет после предобработки данных и удаления неинформативных для классификации признаков стал содержать 86 признаков без учета классификационного. После объединения датасетов в один количество экземпляров составило 255312, большая часть из которых относится к классу «без дефектов» – 240579, меньшее количество данных имеет метку «с дефектами» – 14733.

Для решения проблемы несбалансированных классов был применен метод SMOTE. Таким образом, количество элементов миноритарного класса стало равно количеству элементов мажоритарного класса, а именно 240579. После чего конечный размер датасета составил 481158 экземпляров.

Данный датасет имеет достаточно большое количество признаков с учетом особенностей исследуемой предметной области. Однако не всегда большее количество признаков в обучающих данных положительно влияет на процесс обучения модели. В ходе проведения экспериментов планируется определить наименее важные признаки и сократить размерность датасета для лучшего прогноза модели.

Проведение экспериментов

Для сравнения эффективности и анализа модификаций алгоритма необходимо провести обучение модели без их применения. Так, в *Эксперименте №1* обучение XGBoost проходило с использованием заданных по умолчанию значений гиперпараметров на предобработанном датасете. Несмотря на отсутствие применения каких-либо способов улучшения, модель показала высокие результаты по всем метрикам оценки качества модели:

- точность – 0,9046;
- полнота – 0,9075;
- F1-мера – 0,9065.

Время, потраченное на обучение, составило 8,183 секунды, на предсказание – 0,173 секунды.

Перед проведением *Эксперимента №2* необходимо определить важность признаков в используемом датасете. Для данной цели были применены встроенные инструменты библиотеки алгоритма XGBoost: `weight`, `gain`, `cover`, `total_gain`, `total_cover`. Наиболее информативными и значимыми показателями являются `gain` и `weight`. Опираясь на данные значения для каждого признака, стоит проводить сокращение размерности датасета для исключения признаков, оказывающих негативное влияние на прогностическую способность модели.

Исходя из полученных результатов показателей важности признаков, было отобрано 23 наименее значимых признака в датасете, например, `Multithreading Rules`, `Performance Rules`, `Clone Metric Rules` и другие. Данные признаки были исключены из набора данных, на основе которого был создан новый датасет сокращенной размерности.

Во втором эксперименте модель показала следующие результаты:

- точность – 0,9047;
- полнота – 0,9075;
- F1-мера – 0,9066.

Время, потраченное на обучение, составило 6,433 секунды, на предсказание – 0,181 секунды.

Проведение *Эксперимента №3* заключается в оптимизации гиперпараметров обучения на первоначальном датасете. В таблице 1 представлен список настраиваемых гиперпараметров, диапазон поиска значений и лучшее найденное значение.

Таблица 1

Настройка гиперпараметров для обучения XGBoost

Гиперпараметр	Диапазон значений	Найденное значение
<code>learning_rate</code>	0.001–0.1	0.0924
<code>n_estimators</code>	500–10000, шаг=500	5000
<code>max_depth</code>	8–64	48
<code>subsample</code>	0–1	0.9645
<code>min_child_weight</code>	4–32	4
<code>alpha</code>	0–32	2
<code>lambda</code>	0–32	15

Обучение с использованием оптимизации показало следующие результаты:

- точность – 0,9756;
- полнота – 0,9758;
- F1-мера – 0,9758.

Время, потраченное на обучение, составило 863,293 секунды, на предсказание – 59,244 секунд.

Для проведения *Эксперимента №4* была проведена оптимизация гиперпараметров на датасете сокращенной размерности. Используемые гиперпараметры и диапазон значений такие же, как и в третьем эксперименте. В таблице 2 содержатся значения по настройке гиперпараметров.

Настройка гиперпараметров обучения XGBoost на датасете сокращенной размерности

Гиперпараметр	Диапазон значений	Найденное значение
learning_rate	0.001-0.1	0.0762
n_estimators	500-10000, шаг=500	6000
max_depth	8-64	32
subsample	0-1	0.8898
min_child_weight	4-32	4
alpha	0-32	0
lambda	0-32	10

Результатами обучения модели на производном датасете с оптимизированными гиперпараметрами стали следующие значения:

- точность – 0,9761;
- полнота – 0,9761;
- F1-мера – 0,9762.

Время, потраченное на обучение, составило 1415,249 секунды, на предсказание – 94,04 секунды.

Эксперимент №5 проводился с использованием кросс-валидации на первичном датасете, то есть на модели, полученной в первом эксперименте. Датасет был разделен на 7 частей, соответственно обучение проводилось в 7 итераций. Значения метрик оценки качества модели по результату эксперимента следующие:

- точность – 0,8245;
- полнота – 0,9044;
- F1-мера – 0,8597.

Время, потраченное на обучение, составило 330,891 секунды.

Эксперимент №6 проводился с использованием кросс-валидации на производном датасете, а именно на модели, полученной во втором эксперименте. Как и в пятом эксперименте датасет был разделен на 7 частей. Значения метрик оценки качества модели в данном эксперименте следующие:

- точность – 0,8205;
- полнота – 0,9043;
- F1-мера – 0,8574.

Время, потраченное на обучение, составило 270,33 секунды.

Эксперимент №7 проводился с использованием кросс-валидации на первичном датасете с оптимизированными (на модели, полученной в третьем эксперименте). Обучение происходило в 7 итераций. Значения метрик оценки качества модели в данном эксперименте следующие:

- точность – 0,9106;
- полнота – 0,9736;
- F1-мера – 0,9394.

Время, потраченное на обучение, составило 28861,608 секунды.

В последнем *Эксперименте №8* были применены все рассмотренные ранее варианты модификации XGBoost, то есть обучение модели, полученной в четвертом эксперименте, с применением кросс-валидации. Обучение происходило в 7 итераций, как и в предыдущих экспериментах. Значения метрик оценки качества модели в данном эксперименте следующие:

- точность – 0,9136;
- полнота – 0,9743;
- F1-мера – 0,9415.

Время, потраченное на обучение, составило 20617,99 секунды.

Анализ полученных результатов

Результаты проведенных экспериментов представлены в виде столбчатой диаграммы на рис. 1.

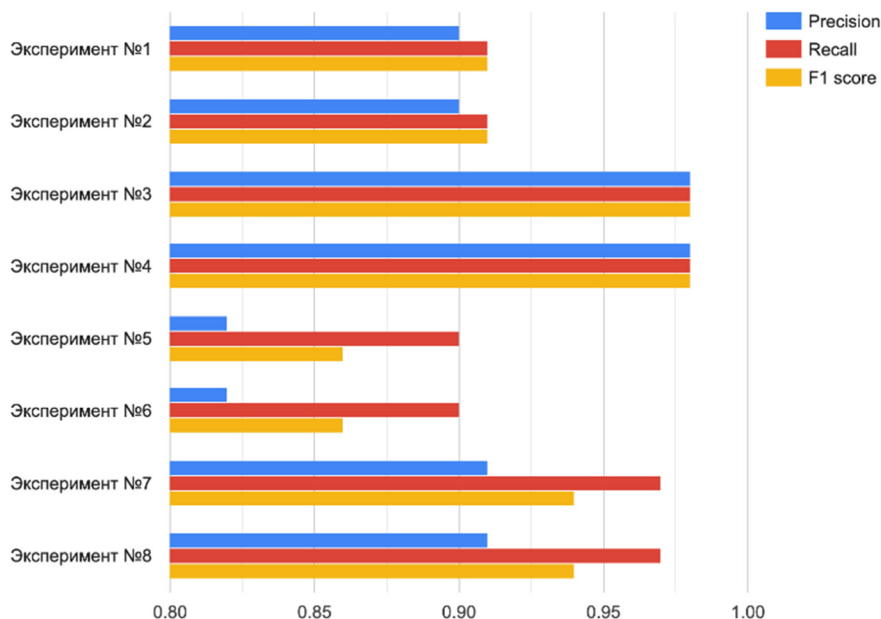


Рис. 1. Сравнение результатов моделей в проведенных экспериментах

Как видно из диаграммы самые низкие значения метрик оценки эффективности моделей имеют модели, обученные в *Экспериментах №5 и №6*, где применяемыми модификациями являлись кросс-валидация (в обоих случаях) и сокращение размерности датасета (в *Эксперименте №6*). Очевидно применение кросс-валидации не оправдывает используемых ресурсов: для расчета показателей по всем метрикам оценки моделей необходимо трижды обучать модель, так как в процессе одного обучения возможен расчет только по одной метрике. Помимо этого, результаты моделей намного ниже по сравнению с моделью, обученной без применения модификаций, что может быть связано с недостаточным количеством данных для обучения в ходе итерации, а именно 1/7 от первоначального датасета.

Результаты *Экспериментов №7 и №8*, в которых также была применена кросс-валидация, заметно лучше, чем результаты двух предыдущих экспериментов. Объяснением подобного скачка показателей метрик оценки моделей является применение оптимизации гиперпараметров. Однако, несмотря на лучшие значения показателей с первыми двумя экспериментами, значения точности F-1 меры для обеих моделей ненамного увеличились, из чего можно сделать вывод, что комбинации применяемых в данных экспериментах модификаций также не принесли существенное улучшение в прогнозе моделей. Кроме того, в данных экспериментах было затрачено больше всего времени на обучение моделей с учетом обучения для каждой метрики оценки: в *Эксперименте №7* более 8 часов, в *Эксперименте №8* около 6 часов – что в значительной степени превышает время обучения в других экспериментах.

В *Эксперименте №2* модель была обучена на датасете сокращенной размерности. Значения точности и F1-меры всего на одну тысячную больше значений, полученных в *Эксперименте №1*, значение полноты не изменилось. Также сократилось время обучения за счет уменьшения количества признаков. Тем не менее, данная модификация, примененная самостоятельно, не несет значительных изменений в эффективности работы модели.

Лучшие результаты были получены в *Эксперименте №3 и №4*, в которых применялась оптимизация гиперпараметров и сокращение размерности датасета (в *Эксперименте №4*). Очевидно, что *оптимизация гиперпараметров является наиболее эффективным способом модификации алгоритма XGBoost*, несмотря на ресурсозатратность поиска оптимальных значений гиперпараметров. Подобранные значения гиперпараметров в обоих случаях можно считать подходящими с учетом особенностей обучения на имеющихся датасетах. Так, при низкой скорости обучения выбрано больше число деревьев решений в составе ансамбля. В *Эксперименте №3* значение регуляризаций подобрано следующим образом: L1 – 2, L2 – 15 – что говорит об использовании датасета с относительно небольшим набором незначительных признаков, но с довольно большим числом связанных признаков.

В *Эксперименте №4* значение L1-регуляризации определено как 0, то есть данный вид регуляризации не применялся, что можно объяснить обучением на датасете, в котором неважные признаки были исключены, соответственно применение данного вида регуляризации является избыточным. Значение L2-регуляризации равно 10, что говорит о наличии коррелированных признаков в датасете и необходимости сохранения всех признаков для участия в делении деревьев.

Метрики оценки качества модели выше на тысячные доли у модели, обученной в *Эксперименте №4*, чем у модели в *Эксперименте №3*. Однако время обучения и предсказания также больше, что можно объяснить большим количеством деревьев в составе ансамбля.

По итогу обучения разных моделей лучшие результаты по всем метрикам оценки качества модели имеет модель, полученная в *Эксперименте №4*.

Заключение

По результатам экспериментов кросс-валидация оказалась неподходящим способом модификации алгоритма из-за сравнительно низких значений метрик и обучения для каждой метрики оценки модели в отдельности. Напротив, лучшие результаты показала модель XGBoost, обученная с применением оптимизированных гиперпараметров на датасете сокращенной размерности. Корректно было подобрано значение для параметра L2-регуляризации, в соответствии с которым были найдены значения других параметров, касающихся архитектуры ансамбля, скорости обучения, размера подвыборки для дерева. Также незначительно данной модели уступает по показателям метрик качества модель, обученная только с применением оптимизированных гиперпараметров. Для данной модели успешно были подобраны значения обоих видов регуляризации: использование L1-регуляризации оправдано в случае обучения на исходном датасете, содержащем немалое количество признаков, ухудшающих прогностическую способность модели.

Подводя итог, лучшей модификацией алгоритма XGBoost для обучения моделей на использованном датасете является совместное применение оптимизации рассмотренных гиперпараметров и датасета сокращенной размерности (без наименее значимых для прогноза признаков).

Литература

1. XGBoost Documentation [Электронный ресурс]: // XGBoost Documentation. Режим доступа: <https://xgboost.readthedocs.io/en/latest/index.html>, свободный. – Загл. с экрана. (дата обращения: 20.08.2025);
2. API Reference [Электронный ресурс]: // scikit-learn. Режим доступа: <https://scikit-learn.org/stable/api/index.html>, свободный. – Загл. с экрана. (дата обращения: 15.04.2025);
3. Software Metrics for Software Defect Prediction Dataset [Электронный ресурс]: // figshare. Режим доступа: https://figshare.com/articles/dataset/Software_Metrics_for_Software_Defect_Prediction_Dataset/21150994, свободный. – Загл. с экрана. (дата обращения: 20.10.2025).
4. Фатхулин Т. Д., Юдин А. Д. Методики оптимизации загрузки изображений в web-приложениях // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 1. С. 105-110. EDN TXTWFG.
5. Фатхулин Т. Д., Фатхулина Г. Г., Рахматова А. А. Интеграция технологии больших языковых моделей в образовательный процесс высшей школы // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 2. С. 107-110. EDN FOGQPZ.
6. Киреев А. А., Фатхулин Т. Д. Анализ средств автоматизированного выбора конфигурации сети // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 1. С. 15-19. EDN ETSHKC.
7. Фатхулин Т. Д., Чепенко К. А. Анализ технологий обнаружения дефектов фасадов зданий // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 1. С. 78-82. EDN BYMERU.
8. Леохин Ю. Л., Фатхулин Т. Д., Кожанов М. С. Анализ и исследование применения нейросетевых технологий для генерации программного кода // Вестник Рязанского государственного радиотехнического университета. 2024. № 87. С. 41-53. DOI 10.21667/1995-4565-2024-87-41-53. EDN HKEOFX.
9. Леохин Ю. Л., Фатхулин Т. Д., Ментус М. В. Разработка и применение методов распознавания зашумленных аудиофайлов посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2024. № 88. С. 65-73. DOI 10.21667/1995-4565-2024-88-65-73. EDN NMXASI.
10. Мяlicheva А. А., Фатхулин Т. Д. Анализ методов машинного обучения для прогнозирования дефектов в исходном коде // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 16-19. EDN IVJCFZ.

11. *Маслов К. В., Фатхулин Т. Д., Иванов Д. А.* Анализ технологий автоматизации бизнес-процессов и разработки программного обеспечения с использованием low-code платформ // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 1. С. 6-11. EDN HDBOYM.
12. *Фатхулин Т. Д., Исаев А. В.* Анализ моделей arima и lstm, используемых для прогнозирования криптовалют и определения портфеля инвестиций // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 20-25. EDN ODWOPA.
13. *Леохин Ю. Л., Фатхулин Т. Д.* Разработка методов и алгоритма формализации текстового запроса к онлайн-сервисам, генерирующим изображения посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2023. № 85. С. 82-95. DOI 10.21667/1995-4565-2023-85-82-95. EDN PZWYZV.
14. *Фатхулин Т. Д., Лушин Е. А.* Анализ развития автоматической генерации кода для web-сервисов // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2023. № 1. С. 128-132. EDN JUEGXP.
15. *Митрофанов А. О., Степанов М. Н., Фатхулин Т. Д.* Анализ нейросетевых методов генерации изображения по текстовому запросу // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2022. № 1. С. 19-23. EDN CWRLQA.
16. *Фатхулин Т. Д., Хорикова С. Г., Щитов В. М.* Анализ ключевых особенностей технологии программно-конфигурируемых оптических сетей (SDON) // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2021. № 1. С. 29-34. EDN SMTDAF.
17. *Simon J.D. Prince, Understanding Deep Learning.* MIT Press, 2023.
18. *Вишневецкий В. М., Леохин Ю. Л., Фатхулин Т. Д., Занегин А. В.* Методы машинного обучения в решении задачи прогнозирования спроса на отдельные виды товаров // Т-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 10. С. 34-43. DOI 10.36724/2072-8735-2024-18-10-34-43. EDN COBEAG.
19. *Леохин Ю. Л., Фатхулин Т. Д., Маслов К. В.* Разработка методов системного анализа бизнес-процессов в банковской сфере для принятия решений о кредитовании различных организаций // Научные технологии в космических исследованиях Земли. 2025. Т. 17, № 5. С. 59-71. DOI 10.36724/2409-5419-2025-17-5-59-71. EDN VXBFTN.
20. *Леохин Ю. Л., Фатхулин Т. Д., Занегин А. В.* Модификация метода градиентного усиления для прогнозирования спроса на отдельные виды товаров // Научные технологии в космических исследованиях Земли. 2025. Т. 17, № 2. С. 32-41. DOI 10.36724/2409-5419-2025-17-2-32-41. EDN PNUPKY.
21. *Леохин Ю. Л., Дымкова С. С., Фатхулин Т. Д.* Методы машинного обучения в прикладных задачах прогнозирования динамично изменяющихся данных // Т-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 8. С. 49-63. DOI 10.36724/2072-8735-2025-19-8-49-63. EDN ULVCHG.
22. *Leokhin Yu. L., Dymkova S. S., Fatkhulin T. D.* Research and development of image improvement tools // Т-Comm: Телекоммуникации и транспорт. 2025. Vol. 19, No. 4, pp. 45-56. DOI 10.36724/2072-8735-2025-19-4-45-56. EDN FUINEN.
23. *Леохин Ю. Л., Дымкова С. С., Фатхулин Т. Д., Зозуля И. С.* Методы и алгоритмы интеллектуальной поддержки принятия управленческих решений в организационных системах торговых компаний // Т-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 12. С. 44-50. DOI 10.36724/2072-8735-2025-19-12-44-50. EDN XXFTQJ.