

На правах рукописи



ДУРОВА МАРИЯ АЛЕКСАНДРОВНА

**ИССЛЕДОВАНИЕ ГИБРИДНЫХ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ
В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ С ИСПОЛЬЗОВА-
НИЕМ ФАКТОГРАФИЧЕСКОГО АНАЛИЗА ТЕКСТА**

Специальность 2.3.5 – Математическое и программное обеспечение вычислитель-
ных систем, комплексов и компьютерных сетей

**Автореферат
диссертации на соискание ученой степени
кандидата технических наук**

Москва – 2026

Работа выполнена на кафедре вычислительных машин, систем и сетей федерального государственного бюджетного образовательного учреждения высшего образования «Национальный исследовательский университет «МЭИ»

Научный руководитель: **Зейн Али Нажиевич**

кандидат технических наук, доцент,
доцент кафедры вычислительных машин, систем и сетей ФГБОУ ВО «НИУ «МЭИ»

Официальные

оппоненты:

Артём Владимирович Горчаков

кандидат технических наук, доцент кафедры корпоративных информационных систем ФГБОУ ВО «МИРЭА – Российский технологический университет»

Ведущая организация:

Федеральное государственное автономное учреждение "Федеральный научный центр Научно-исследовательский институт системных исследований Национального исследовательского центра "Курчатовский институт"

Защита диссертации состоится «»2026 г. в часов 00 мин. на заседании диссертационного совета МЭИ.105 при ФГБОУ ВО НИУ «МЭИ» по адресу: 111250, г. Москва, ул. Красноказарменная, д. 13, стр. 5, ауд. М-805.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВО «НИУ «МЭИ».

Автореферат разослан _____ 2026 г.

Ученый секретарь диссертационного совета

МЭИ.105 к.т.н., доцент

А.М. Чернецов

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования. В системах поддержки принятия решений (СППР), работающих с текстом, под знаниями понимается структурированная информация, пригодная для логического вывода и объяснения решений. СППР, работающие с такой информацией, как правило, функционируют в рамках динамической среды. В данном контексте "динамически изменяющаяся среда" означает, что предметная область, с которой работает СППР, постоянно эволюционирует: появляются новые термины, формируются новые категории и связи. Неструктурированный текст – это носитель семантики, которая не фиксирована заранее, и поэтому требует адаптивных моделей знаний, в отличие от структурированных данных с фиксированной схемой.

Традиционно модели знаний в СППР строились на жёстко заданных экспертных правилах «если–то». Однако в динамической среде – где смысл терминов и их взаимосвязи со временем меняются – такой подход оказывается недостаточным. Таким образом, возникает задача создания системы, способной самостоятельно дообучаться и генерировать новые правила, когда заранее заданных знаний недостаточно.

В рамках этого направления ведется активная разработка методов и программных средств построения гибридных интеллектуальных информационных систем (ГИИС). Как правило, этот термин используется для обозначения систем, реализующих интеграцию методов экспертного обучения, нечёткой логики, эволюционных алгоритмов или иных подходов, но при этом не существует единого ответа на вопрос, каким образом реализовать принцип гибридности. Одним из наиболее активно развивающихся в последние годы подходом является использование в качестве основы для такой интеграции общей модели знаний.

Процесс извлечения из текста проверяемых фактов и их структурирования для последующего анализа называется фактографическим анализом, широко используемым методом интеллектуального анализа данных, извлекаемых из текстового потока. При этом методология построения набора правил, на основе которых будет осуществляться поиск фактографических правил, является узким местом любой системы фактографического поиска. Эта задача традиционно решается экспертными методами, однако в случаях динамической среды это сопряжено с высокими издержками. Перспективным улучшением этого подхода представляется описание знаний предметной области в виде формальной модели.

Степень разработанности темы. Следует отметить, что ГИИС относятся к классу сложных информационных динамических систем – динамических интеллектуальных систем (ДИС), основанных на использовании методов искусственного интеллекта и моделировании процессов принятия решений. Вопросам представления знаний и моделирования динамики в ГИИС как подкласса ДИС посвящены работы многих отечественных и зарубежных исследова-

телей: Еремеева А.П., Королёва Ю.И., Ревункова Г.И., Ярушкиной Н.Г., Колесникова А.В., Cruse H. и др. В области моделирования сложных зависимостей в распределенных системах существенный вклад был внесен Вагиным В.Н. и Топорковым В.В., чьи исследования по интеллектуальному анализу данных и оптимизации ресурсов в гетерогенных средах заложили основу для разработки формализованных подходов к управлению распределенными вычислениями. Эти работы подчеркивают роль графовых структур как инструментов для анализа связей в информационных структурах, что позволяет эффективно моделировать динамические взаимодействия между компонентами систем. В этом контексте метаграфы, предложенные A. Basu и R.W. Blanning, представляют собой расширенный аппарат для формализации сложных зависимостей. В настоящее время устойчивый интерес проявляется к модели представления знаний в ГИИС на основе метаграфов, где эта структура интегрируется с методами машинного обучения для повышения адаптивности и объяснимости результатов. Этот подход предлагает информационную структуру, в которой связи между понятиями могут быть как статическими – заданными вручную правилами, динамическими – метриками, извлечёнными из данных, процессуальными – правилами обработки, управляющими преобразованием знаний. Однако многие принципы практической реализации ГИИС на основе этой модели были сформулированы теоретически, но не реализованы на практике, и таким образом реальная их применимость остается недоказанной.

Цель и задачи работы: исследовать гибридные методы машинного обучения в системах поддержки принятия решений и разработать архитектуру гибридной СППР, обеспечивающую интеграцию методов машинного обучения и экспертных правил при работе с неструктурированными текстовыми данными в условиях динамической семантики и прагматики. Для достижения поставленной цели требуется решить следующие основные задачи:

1. Проанализировать существующие подходы к построению гибридных систем на основе машинного обучения.
2. Разработать математическое описание информационного элемента метаграфа (ИЭМ) и доказать корректность базовых операций над метаграфом.
3. Разработать архитектуру СППР, основанную на метаграфах для интеграции методов машинного обучения.
4. Реализовать прототип механизма формирования правил, позволяющий системе адаптироваться к изменяющейся среде без обучения системы с нуля.
5. Провести сравнительный анализ эффективности предложенной архитектуры с традиционным подходом к представлению знаний, таким как онтология.

Объектом исследования являются гибридные интеллектуальные информационные системы для работы с неструктурированными текстовыми данными.

Предметом исследования являются методы описания знаний и процессов в ГИИС, основанных на метаграфовой модели.

Научная новизна:

Научная новизна работы определяется разработкой методов и моделей для ГИИС в динамических текстовых средах, что включает:

1. Новый метод использования метаграфовой модели, гарантирующий согласованность структур при линейной сложности операций.
2. Метод описания ИЭМ с определением связей и ограничений для обеспечения целостности и корректности в ГИИС.
3. Архитектура ГИИС, объединяющая фактографический анализ, семантическое расширение и ранжирование в метаграфе.
4. Методика агентов для реализации модели, сокращающая фазы обработки задач ранжирования.

Теоретическая и практическая значимость работы. В диссертации предложена формализация метаграфовой модели как информационной структуры, что позволит в дальнейших исследованиях использовать ее при разработке ГИИС. Доказана ее применимость для построения систем, способных к принятию объяснимых решений, что является одним из способов обеспечения доверия к системам ИИ согласно с ГОСТ Р 59276– 2020.

Принципы построения ГИИС, предлагаемые в работе, могут быть применены для эффективной работы с неструктурированными текстовыми данными в условиях постоянных изменений предметной области, например в задачах динамического мониторинга новостных потоков, маршрутизации документооборота, анализа рынка труда и так далее.

В диссертации разработаны методы работы с метаграфовой моделью, обеспечивающие согласованность и непротиворечивость информационных структур. Физическое хранение метаграфа в системе обеспечено с помощью графовой базы данных Neo4j Database Enterprise Edition 5, запросы к которой выполняются на языке запросов Cypher. Так как на данный момент не существует программных средств для хранения метаграфовой модели знаний напрямую, она представлена в проекции на четырехдольный граф. В свою очередь логика метаграфовых агентов реализована с помощью языка Python 3, а их отладка производилась в среде PyCharm 2023.2.2. Система построена как холоническая многоагентная система, где все компоненты – от данных до процессов – представлены в виде ИЭМ. Это обеспечивает единую модель знаний, в которой данные, процессы и правила существуют на одном уровне. Были реализованы два прототипа системы – для маршрутизации входящей электронной корреспонденции и для аналитики рынка труда в IT-сфере. Таким образом, в диссертационной работе предложено их практическое применение.

Методология и методы исследования. В основе диссертационного исследования лежат ZF-теория множеств, формальная логика первого порядка, теория графов, математическая индукция. Для разработки прототипов ГИИС использовались методика применения агентов, а при реализации их логики применялись методы обработки естественного языка (токенизация, лемматизация, шинглирование), алгоритмы сравнения текста (алгоритм шинглов, косинусное сходство, оценка Жаккара, минимальное хеширование, хеширование с учетом локальности), алгоритм полнотекстового поиска BM25F, предобученная модель для генерации векторных представлений Sentence-BERT.

Положения, выносимые на защиту:

1. Разработанный метод применения метаграфовой модели для ГИИС, обеспечивающий согласованность и непротиворечивость информационных структур (п.4 паспорта специальности 2.3.5).
2. Разработанный метод описания объектов, где для каждого ИЭМ описываются его связи с другими объектами метаграфа и формируется список программных ограничений для обеспечения целостности структуры ГИИС и корректности базовых операций в ней (п.4 паспорта специальности 2.3.5).
3. Разработанная архитектура ГИИС в единой метаграфовой модели (п.4 паспорта специальности 2.3.5).
4. Разработанная методика применения агентов, сокращающая количество фаз для задач ранжирования и сортировки (п.5 паспорта специальности 2.3.5).

Степень достоверности и апробация результатов. Основные результаты диссертационного исследования обсуждались на научно-практических конференциях, среди которых: «4th International Youth Conference on Radio Electronics, Electrical and Power Engineering» (Moscow, 2022), «Технологии будущего: VI Международная научно-техническая конференция студентов и аспирантов» (Москва, 2022), «5th International Youth Conference on Radio Electronics, Electrical and Power Engineering» (Moscow, 2023), «Технологии будущего: VII Международная научно-техническая конференция студентов и аспирантов» (Москва, 2023), «6th International Youth Conference on Radio Electronics, Electrical and Power Engineering» (Moscow, 2024), «XXXII Международная научно-техническую конференция студентов и аспирантов «Радиоэлектроника, электротехника и энергетика» (2025 г., Москва)

Реализация результатов. Автором были разработаны и зарегистрированы программы для ЭВМ: «Программное приложение для аналитики рынка труда и рекомендации профессиональных навыков на основе метаграфовой модели» (свидетельство №2025692521), «Программное приложение для внесения, редактирования и обработки настроек программных комплексов для численного моделирования энергетических процессов, версия 2» (свидетельство

№2023686751), «Программное приложение для автоматической маршрутизации входящей корреспонденции на основе гибридной метаграфовой модели» (свидетельство №2025689047). Последняя программа предназначена для сортировки и фильтрации входящих писем корпоративной почты и использована в ООО «ЭНЕРГОРЕМОНТ-2» (акт об использовании научных результатов № 5/21-10-2025), в учебном процессе кафедры Вычислительных машин, систем и сетей ФГБОУ ВО "НИУ "МЭИ" и в АО «Медиаскоп» (заключение об использовании научных результатов № 242/12/25).

Соответствие паспорту научной специальности. Содержание диссертационной работы соответствует направлениям исследований по пунктам № 4 и 5 паспорта специальности 2.3.5. «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: 4. Интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов. 5. Программные системы символьных вычислений.

Структура диссертации. Диссертационная работа состоит из введения, 4 глав, заключения, списка литературы, содержащего 86 наименований использованных источников, и 4 приложения. Объем работы составляет 176 страниц, включая 15 таблиц и 28 рисунков.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе рассматривается обобщенное по многим классическим и современным источникам определение СППР. Согласно этому определению, СППР – это интерактивная информационная система, которая использует данные и модели выбора решений, обеспечивает пользователям доступ к информационным ресурсам и предоставляет им разнообразные возможности в части обработки информации и принятия решений.

Рассматриваются основные проблемы при построении современных СППР. Первая из них – *низкая объяснимость решений* в системах, использующих методы машинного обучения. Методологическая поддержка в современных работах выделяется как отдельная подсистема, что указывает на внутреннюю необъяснимость самой СППР. Ставится задача интеграции объяснения в сам процесс вывода. Вторая проблема – *ограниченность статических моделей знаний*. Необходимость самоорганизации системы в изменяющейся среде подчеркивается в фундаментальных работах по динамическим интеллектуальным системам, однако вопрос пополнения и эволюции знаний и правил в них не рассматривается. Ставится задача реализации механизма самоизменения модели знаний на основе анализа данных. Решение этих двух задач в рамках одной системы представляет собой требование к классу систем, исследуемых в данной работе. Предлагается обозначить этот класс как «СППР с текстовой динамической семантикой и прагматикой» (СППР-ТДСП). СППР-ТДСП является подклассом ДИС, в котором:

1. Семантика (представление знаний) динамична: модель знаний может самостоятельно изменяться (расширяться, уточняться) в процессе работы.
2. Прагматика (процесс принятия решений) динамична: цели и стратегии системы могут адаптироваться на основе обратной связи и анализа среды.
3. Архитектура основана на гибридизации разнородных компонентов в рамках единой информационной структуры.

Факт представляет собой структурированную единицу знания, извлечённую из неструктурированного текста документа. Формально факт f определяется как кортеж:

$$f=(t, s, p, o), \quad (1)$$

где:

t – тип факта ($t \in \{ACTION_REQUEST, ENTITY_RELATION, TOPIC, DEADLINE, PRIORITY, ORGANIZATIONAL_QUERY, \dots\}$);

s – субъект (кто/что является инициатором или носителем);

p – предикат (действие или отношение);

o – объект (на что направлено действие).

Каждый факт сопровождается мерой уверенности $c \in [0,1]$, которая отражает степень семантической согласованности факта с исходным текстом документа и вычисляется на основе семантического сходства.

Требуется поддерживать модель знаний $K(t)$ и функцию обработки f такую, что при поступлении нового письма T_{n+1} выполняется соотношение $K(t+1)=\Omega(K(t), T_{n+1})$, где Ω – оператор обновления модели знаний, реализующий минимальное согласованное расширение текущей модели.

Во второй главе приводится анализ применимости в СППР-ТДСП такой модели представления знаний, как онтология. В качестве примера строится система маршрутизации входящих документов. Документ поступает в систему в виде частично структурированного объекта (поля «тема», «тело», «отправитель»), должен быть определен как относящийся к одной из категорий и назначен к пользователю определенной роли с указанием рекомендуемого действия. Эксперименты проводились на выборке из 80000 экземпляров.

Онтологическая модель – формальное представление знаний, которое описывает концептуальную структуру предметной области, включая объекты (сущности), их свойства, отношения между ними и ограничения, которые на них накладываются. Основная цель онтологии – обеспечить семантическую основу для обмена и интерпретации данных в рамках данной предметной области. Формально она описывается как структура, где X – множество понятий онтологической системы, R – связи между ними, F – набор интерпретирующих функций и методов,

Ax – это набор аксиом, который обеспечивает формирование истинных предикативных выражений, аргументами которых являются понятия, их свойства, отношения, функции и методы онтологии, а R_s – ограничения, определяющие область применения концептуальных структур конкретной предметной области:

$$O = \langle X, R, F, Ax, R_s \rangle \tag{2}$$

Была построена (рис. 1) обобщенная информационная модель системы, основанная на модульной архитектуре. Описание функций и используемых данных каждого модуля представлено в таблице 1.

Данная модель была реализована с применением программных средств для работы с онтологиями (Protégé 5.0, библиотека rdflib v6.3.0), где сама онтология представлена в формате представлена в формате RDF/Turtle). Для оценки пригодности онтологического подхода к построению СППР-ТДСП было проведено экспериментальное исследование, направленное на измерение ключевых метрик: времени обработки, масштабируемости и трудозатрат на модификацию знаний.

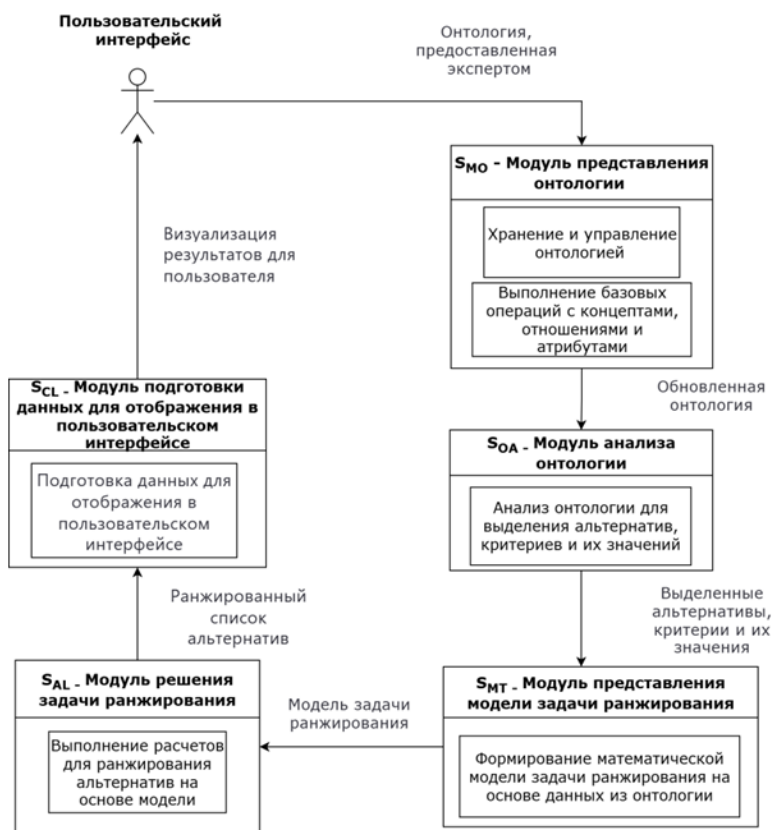


Рис.1 – Информационная модель системы, основанная на модульной архитектуре

Полный цикл обработки документа представлен в таблице 2. Его можно рассматривать как два принципиально разных компонента: ядро принятия решений, выполняющее классификацию, генерацию альтернатив и ранжирование, и механизм анализа онтологии, так называемая машина логического вывода (МЛВ) или reasoner. Он анализирует формализованные знания, проверяет их на логическую непротиворечивость и позволяет автоматически выводить новые факты и классы из заданных правил и аксиом.

Таблица 1 – Элементы информационной системы на основе онтологии

Название	Функция	Входные данные	Выходные данные
----------	---------	----------------	-----------------

S_{MO}	Модуль представления онтологии	Хранение и управление онтологией, выполнение базовых операций (добавление, удаление, изменение концептов, отношений и атрибутов).	Онтология, предоставленная экспертом	Обновленная онтология, доступная для других модулей.
S_{OA}	Модуль анализа онтологии	Анализ онтологии для выделения альтернатив, критериев и их значений	Онтология из модуля S_{MO}	Выделенные альтернативы, критерии и их значения.
S_{MT}	Модуль представления модели задачи ранжирования	Формирование математической модели задачи ранжирования на основе данных из онтологии.	Альтернативы, критерии и их значения из модуля S_{OA} .	Модель задачи ранжирования.
S_{AL}	Модуль решения задачи ранжирования	Выполнение расчетов для ранжирования альтернатив на основе модели	Модель задачи ранжирования из модуля S_{MT}	Ранжированный список альтернатив.
S_{CL}	Модуль подготовки данных для интерфейса	Подготовка данных для отображения в пользовательском интерфейсе.	Результаты ранжирования из модуля S_{AL} .	Визуализация результатов для пользователя.

Таблица 2 – Полный цикл обработки одного документа

Этап	Описание
1. Получение данных	Приём документа.
2. Генерация TTL-представления	Создание временного файла.ttl, содержащего RDF-описание документа.
3. Парсинг TTL-файла	Чтение файла и загрузка его содержимого в отдельный <code>rdflib.Graph()</code> .
4. Объединение графов	Добавление данных документа в основной рабочий граф онтологии.
5. Запуск МЛВ	Применение всех правил вывода OWL RL к полному графу.
6. Классификация типа документа ($F_{classify}$)	Анализ темы и текста для определения категории: запрос, жалоба и т.д.
7. Определение приоритета ($F_{priority}$)	Установка приоритета на основе типа документа (например, запрос \rightarrow 1).
8. Генерация альтернатив (F_{sel})	Формирование множества пар (тип, действие) на основе онтологии и роли пользователя. Например, для запрос и Менеджер: [("запрос", "переслать"), ("запрос", "создать задачу")].
9. Проверка ограничений (R_s)	Проверка, может ли пользователь выполнить действие (например, <code>ex:СоздатьЗадачу</code> , <code>ex:allowedForRole</code> , <code>ex:Менеджер</code>).
10. Оценка критериев (F_{crit})	Расчёт значений по критериям: приоритет, роль, срочность, вес действия.
11. Ранжирование (G_{ra})	Применение правила ранжирования (взвешенная сумма) для упорядочивания альтернатив.
12. Подготовка результата	Форматирование вывода для S_{CL} (номер, тип, действие, оценка).

В ходе первого эксперимента измерялось время обработки нового документа при загрузке в онтологию, содержащую N_{doc} документов, образующих N_{RDF} триплетов ("субъект-предикат-объект"). Помимо общего времени обработки нового документа T_{total} измерялось время $T_{S_{AL}}$ его обработки в модуле S_{AL} , так как именно он является ядром принятия решений в данной архи-

текстуре. В этом эксперименте запуск МЛВ не осуществлялся, то есть система функционировала в статичном режиме, подразумевая, что все нужные правила уже существуют в онтологии. Приведенные значения являются усредненными данными после 15 итераций. Качество принятия решения определяется F_1 -мерой, где истинные значения - выбор рекомендуемого действия – были определены экспертным методом. По результатам эксперимента (таблица 3) видно, что наблюдается линейная зависимость как общего времени, так и времени обработки в модуле S_{AL} .

Таблица 3 - Масштабирование системы в статичном режиме

N_{doc}	$NRDF$	$T_{total\ avg, c}$	$T_{S_{AL}\ avg, c}$	F_1
10	697	0,047	0,023	0,87
100	5197	0,043	0,024	0,86
800	50197	0,054	0,029	0,87
8000	500197	0,056	0,031	0,85
80000	5000197	0,054	0,031	0,88

Второй эксперимент, условия которого приведены в таблице 4, имел своей целью измерить затраты на поддержание целостности онтологии при масштабировании системы в динамическом режиме, то есть с активным МЛВ. Для моделирования реальной нагрузки МЛВ запускался не после каждого документа, а после каждой группы из 100 документов.

Таблица 4 - Масштабирование системы в динамическом режиме ¹⁾

N	$T_{total, c}$	$T_{res, c}$	Доля T_{res} в общем цикле	$T_{avg, c}$	F_1
1	0,605	0,567	94%	0,605	0,92
10	1,390	1,309	94%	0,139	0,91
100	3,904	3,798	97%	0,039	0,93
1000	66,263	59,202	89%	0,066	0,93
5000	1308,510	1273,021	97%	0,262	0,92
50000	26170,200	25123,392	96%	0,523	0,94
80000	78510,600	75370,176	96%	0,981	0,95

Как показывают результаты, первый компонент работает быстро (~7 мс на письмо) и масштабируется линейно. Второй же компонент, несмотря на то, что не участвует напрямую в формировании рекомендаций, доминирует в общем времени выполнения, достигая 98%. Это свидетельствует о том, что основная вычислительная нагрузка в онтологической СППР вызвана не анализом данных, а необходимостью постоянной проверки формальной корректности модели знаний. Кроме того, при активном МЛВ зависимость длительности обработки нового документа от количества документов в онтологии не линейна. Также в ходе исследования было установлено, что, хотя онтологии предоставляют возможность формализовать базовые логические зависимости через аксиомы Ax и интерпретирующие функции F , они не поддерживают вложенность правил, что является критическим недостатком для систем, стремящихся к само-

организации. Правило не может иметь условием другое правило, что делает онтологию неспособной к формированию эмерджентного поведения, где новые правила возникают из взаимодействия существующих.

Как видно из результатов экспериментов, в режиме динамическом режиме показывает высокую точность решения, так как новые документы пополняют базу знаний, однако при этом накладные расходы таковы, что делают систему непригодной к обработке большого потока документов. Таким образом, несмотря на высокую точность системы в динамическом режиме, нелинейный рост времени принятия решения делает онтологию непригодной моделью знаний для СППР-ТДСП, так как способность к дообучению является одним из ключевых требований к системам этого класса.

В третьей главе рассматривается существующая модель метаграфа, и развивается его формальное описание. Метаграф MG определяется как структура (3)

$$MG = \langle V, E, MV, ME \rangle, \quad (3)$$

где: V – множество вершин метаграфа, E – множество рёбер, MV – множество метавершин метаграфа, ME – множество метарёбер.

В отличие от простого графа, где все узлы и рёбра однотипны, метаграф позволяет моделировать сложные системы с множеством уровней взаимодействия с помощью таких структур, как метавершина и метарёбро.

Для решения задачи реализации метаграфа в базе данных используется Информационный Элемент Метаграфа (ИЭМ) – единая структура данных, служащая элементарной строительной единицей для конструирования метаграфов. Идея ИЭМ заключается в том, чтобы унифицировать представление всех элементов метаграфа в единую объектную структуру, которая может быть реализована в любой системе управления базами данных (реляционной, объектной, графовой). Типы и размерности данных зависят от конкретной реализации. Формально ИЭМ представляется в виде кортежа:

$$I = (id, NM, VAL, RL, L, A), \quad (4)$$

где: id – уникальный идентификатор элемента; NM – наименование элемента, VAL – значение элемента; RL – тип элемента: RLV (вершина), $RLMV$ (метавершина), RLE (ребро), $RLME$ (метарёбро); L – множество ссылок на другие ИЭМ; A – множество ссылок на атрибуты.

Таким образом, любая структура метаграфа – от простой вершины до сложной метавершины, содержащей вложенный фрагмент, – может быть построена из набора ИЭМ, связанных друг с другом. Однако существует проблема: хотя некоторые операции над ИЭМ описаны в современных исследованиях, математические гарантии их корректности не обеспечены. В контексте данного исследования для обеспечения строгой математической основы системы утвер-

жений об ИЭМ используется ZF-теория множеств. Универсумы U, N, V, R будут определены как множества, существование которых гарантируется аксиомами ZF: U – множество всех идентификаторов, N – множество наименований, V – множество значений, $R = \{RLV, RLMV, RLME, RLE\}$ – множество типов элементов, так называемых ролей ИЭМ. Кортеж I будет формализован через множества: $id \in U, NM \in N, VA \in V, RL \in R, L \subseteq U$ – конечное множество ссылок, $A \subseteq U$ – конечное множество ссылок на атрибуты.

Утверждения об ИЭМ:

P1. Уникальность идентификатора. Для любых двух различных ИЭМ их идентификаторы также различны:

$$\forall I_1, I_2 \in MG : (I_1 \neq I_2) \Rightarrow (id(I_1) \neq id(I_2)) \quad (5)$$

Это утверждение устанавливает фундаментальное свойство однозначной идентификации. Оно гарантирует, что отображение из множества всех ИЭМ в множество идентификаторов является инъективным (один к одному). Это позволяет использовать id как первичный ключ в любой реализации системы.

P2. Целостность ссылок. Каждая ссылка, содержащаяся в множестве $L(I)$ любого ИЭМ I , указывает на существующий ИЭМ I' в метаграфе:

$$\forall I \in MG, \forall id' \in L(I) : \exists I' \in MG : id(I') = id' \quad (6)$$

Это утверждение обеспечивает структурную целостность метаграфа. Оно гарантирует, что все ссылки являются действительными и не указывают на несуществующие элементы. Это аналогично понятию целостности внешнего ключа в реляционных базах данных.

P3. Ролевая принадлежность. Роль любого ИЭМ обязательно принадлежит одному из четырёх допустимых типов: вершина, метавершина, ребро, метаребро.

$$\forall I \in MG : RL(I) \in \{RLV, RLMV, RLE, RLME\} \quad (7)$$

Это утверждение определяет типовую систему для ИЭМ. Оно устанавливает, что роль – это дискретный классификатор, который жёстко ограничивает возможные значения. Это позволяет системе однозначно интерпретировать каждый ИЭМ: как базовый элемент данных (RLV), как сложный объект (RLMV, RLME), или как связь (RLE).

P4. Атрибутивная целостность. Если id' – это идентификатор, на который ссылается атрибут $A(I)$ ИЭМ I , то существует ИЭМ I' в метаграфе, имеющий роль RLV:

$$\forall I \in MG : (id' \in A(I)) \Rightarrow (\exists I' \in MG : (id(I') = id') \wedge (RL(I') = RLV)) \quad (8)$$

Это утверждение определяет семантику атрибутов в системе. Оно гарантирует, что атрибут – это не любая сущность, а только ИЭМ с ролью RLV. Это упрощает интерпретацию: все атрибуты – это "значения", а не сложные объекты или связи.

Утверждение также гарантирует, что ссылки в $A(I)$ указывают на реальные элементы, что обеспечивает целостность данных.

Р5. Связность ребра. Если ИЭМ имеет роль RLE (ребро), то он должен содержать ровно две ссылки в своём множестве $L(I)$.

$$\forall I \in MG : (RL(I)=RLE) \Rightarrow (|L(I)|=2) \quad (9)$$

Это утверждение определяет структуру бинарной связи. RLE интерпретируется как направленное или ненаправленное ребро, соединяющее ровно два ИЭМ. Это соответствует классическому пониманию связи в графе.

Утверждение исключает возможность создания метарёбер с помощью RLE, что сохраняет чёткое разделение между ролями: *RLE* – для бинарных связей, а сложные структуры – через *RLME*.

Р6. Структурная целостность метавершины. Если метавершина не содержит ни одной ссылки, то такая метавершина не может существовать в метаграфе:

$$\forall I \in MG : (RL(I)=RLMV \wedge L(I)=\emptyset) \Rightarrow \neg \exists I' \in MG : (id(I')=id(I)) \quad (10)$$

Метавершина – это не просто узел, а контейнер для вложенного фрагмента метаграфа. Это отражает суть метавершины как носителя эмерджентности. Данное утверждение формализует это, запрещая существование "пустых" метавершин, которые не содержат вложенного фрагмента. Пустая метавершина может существовать в метаграфе только как временный элемент, предназначенный для последующего заполнения. Такая метавершина должна быть помечена атрибутом *status="pending"*. При реализации утверждение Р6 подразумевает не просто наличие ссылок, а наличие ссылок и отсутствие атрибута *status="pending"*.

Р7. Утверждение эмерджентности (рекурсия). Пусть I – ИЭМ, тогда $id(I)$ возвращает его id , а $L(I_{parent})$ – множество связей с I внутри MG . Тогда составные элементы метаграфа, такие как метаребра и метавершины, могут содержать в себе другие элементы того же типа:

$$\forall I_{parent} \in MG : (RL(I_{parent})=RLMV \vee RL(I_{parent})=RLME) \Rightarrow (\exists I_{child} \in MG : (id(I_{child}) \in L(I_{parent})) \wedge (RL(I_{child})=RLMV \vee RL(I_{child})=RLME)) \quad (11)$$

Кроме того, были определены некоторые ограничения на структуру метаребер:

Ограничение М1: Все ребра и метаребра во вложенном фрагменте метаребра должны быть направленными ($eo = true$).

Цель ограничения - гарантировать, что внутренний процесс имеет четкое направление выполнения (последовательность операций).

Ограничение М2: Метаребро (me_i) должно иметь обе связанные точки: исходную (v_S) и конечную (v_E).

Цель ограничения - обеспечить семантическую корректность и завершённость процесса, без его соблюдения метаребро не может быть интерпретировано как направленный процесс. Это частный случай утверждения P2 (Целостность ссылок). Система должна гарантировать, что v_S и v_E существуют в MG , с тем уточнением, что точками могут являться вершины либо метавершины.

Далее в главе были приведены основные операции над метаграфом и исследована их масштабируемость. В ходе экспериментов было установлено, что установка индекса на уникальный идентификатор может свести сложность выполнения операции создания ИЭМ с $O(n)$ до $O(m)+O(k)$, где n – число узлов в БД, m – число атрибутов каждого ИЭМ, k – количество связей ИЭМ с другими ИЭМ (рис.2 и 3):

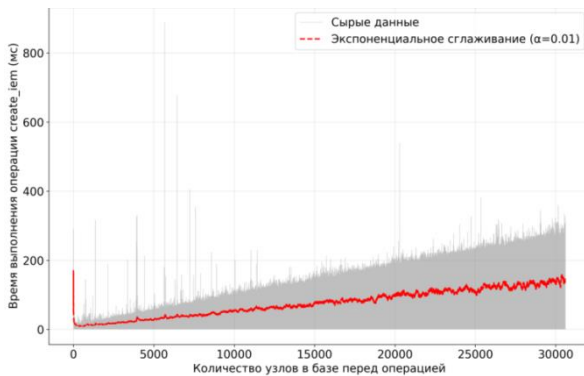


Рис.2 – Зависимость времени создания ИЭМ от размера метаграфа (без индекса по id)

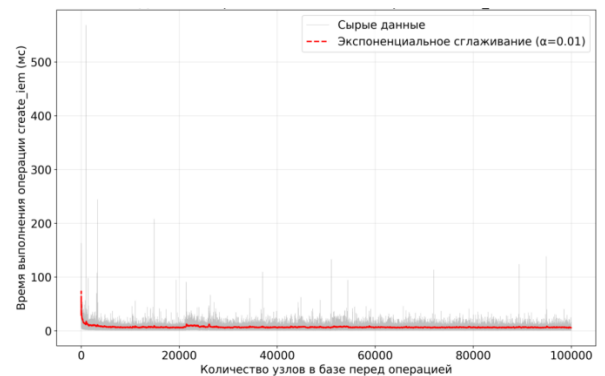


Рис.3 – Зависимость создания ИЭМ от размера метаграфа (без индекса по id)

Было установлено также, что сложность создания и расширения метавершины составляет $O(p)$, где p – количество добавляемых ИЭМ, вне зависимости от роли.

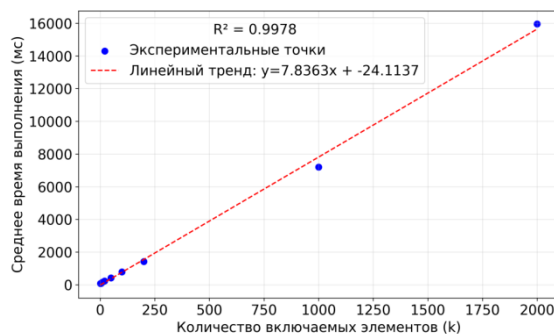


Рис.4 – Зависимость времени создания метавершины от количества включаемых ИЭМ

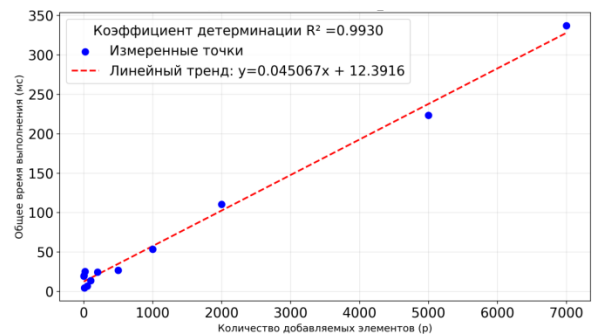


Рис.5 – Зависимость скорости расширения метавершины от количества добавляемых ИЭМ

В рамках главы предложена система утверждений об информационном элементе метаграфа (ИЭМ), обеспечивающая математически строгую основу для представления знаний и гарантирующая корректность базовых операций над метаграфовой структурой. Эта формализация

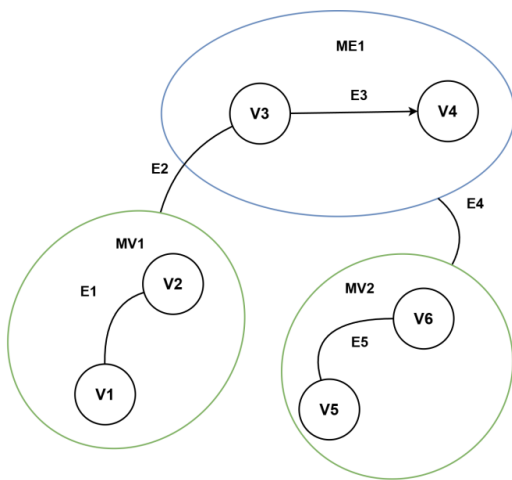
напрямую реализует первый пункт научной новизны и позволяет обеспечить согласованность и непротиворечивость информационных структур при их динамическом изменении.

В четвертой главе приводится описание программной реализации прототипов СППР-ТДСП на основе метаграфовой модели. В качестве физической модели хранилища данных была выбрана графовая база данных. В такой модели реализуется отображение метаграфа в плоский четырёхдольный граф:

$$FG = \langle FG^{VERT}, FG^{EDGE} \rangle, \quad (12)$$

$$FG^{VERT} = \langle FG^V, FG^E, FG^{MV}, FG^{ME} \rangle,$$

$$\langle FG^V \leftrightarrow MG^V \rangle, \langle FG^E \leftrightarrow MG^E \rangle, \langle FG^{MV} \leftrightarrow MG^{MV} \rangle, \langle FG^{ME} \leftrightarrow MG^{ME} \rangle,$$



где FG^V , FG^E , FG^{MV} и FG^{ME} – подмножества вершин графа, изоморфных множествам вершин, рёбер, метавершин и метарёбер метаграфа, а FG^{EDGE} – множество рёбер графа, осуществляющие технические связи между его долями. На рисунке 6 изображен фрагмент метаграфа, а рисунок 7 показывает его же в проекции на четырёхдольный плоский граф.

Рис.6 Исходный фрагмент метаграфа

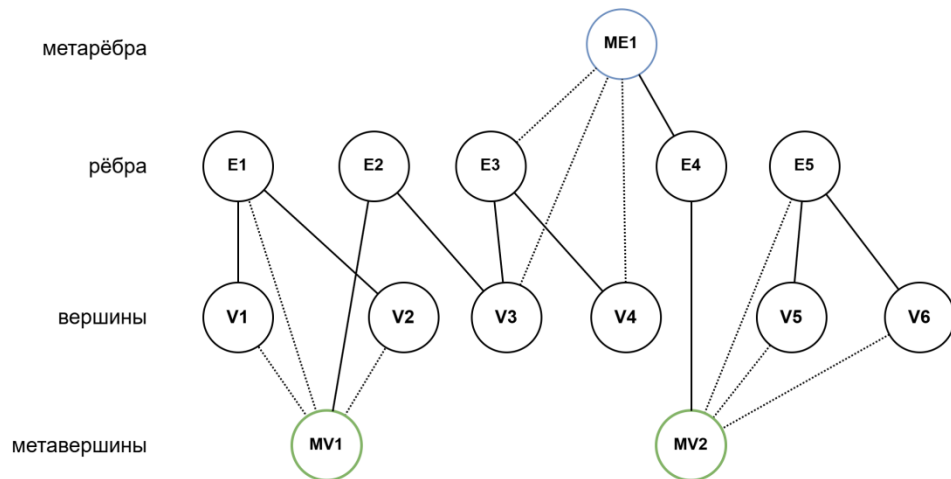


Рис.7 Выровненный фрагмент метаграфа

Помимо прототипа, решающего задачу маршрутизации документов, аналогичную описанной во второй главе, был разработан прототип системы стратегических рекомендаций на основе анализа рынка труда. Она собирает данные вакансий в сфере ИТ, выделяет ключевые навыки, сопоставляет их между собой и предлагает список дополнительных навыков, релевантных каждой вакансии, а также определяет предположительный диапазон заработной платы для вакансий, где она не была указана.

В разработке обоих прототипов были использованы принципы холонической много-агентной архитектуры, где агенты организованы в иерархическую структуру. Каждый агент в системе реализован как специализированная компонента, взаимодействующая с единой моделью знаний напрямую, так как логика его работы сама встроена в эту модель в виде метаребра. В таблице 5 приведено использование каждого агента в обеих системах.

Таблица 5 - Масштабирование системы в динамическом режиме

Агент	Прототип 1	Прототип 2
AgNLP	Извлекает подпоследовательности слов из текста документа.	Извлекает ключевые навыки из описания вакансии.
AgSrch	Ищет похожие документы.	-
AgML	Ранжирует документы по релевантности.	Поиск схожих вакансий для предсказания не указанных зарплат.
AgSemanticExpander	Расширяет подпоследовательности слов семантически.	Используется для расширения навыков (например, "Python" → "FastAPI").
AgUI	Собирает обратную связь от пользователя.	(Не используется, если система работает в автономном режиме).
AgRule	Генерирует правила для классификации документов.	Генерирует правила для категоризации навыков и определения высокооплачиваемых вакансий.
AgAdapt	Управляет последовательностью агентов, пропуская этапы при наличии правил.	Управляет последовательностью агентов, пропуская этапы при наличии правил.

Система не просто накапливает опыт, но и формализует его в виде новых знаний-правил. Новое правило может перехватывать поток обработки, позволяя пропустить этапы AgSrch и AgML для уже известных шаблонов, что значительно повышает производительность, что видно из результатов, представленных в таблице 6:

Таблица 6 - Влияние примененных правил на производительность системы

№	Конфигурация	t _c , мс	F ₁
1	Базовая система: AgNLP → AgSem → AgSrch → AgML	250	0,94
2	Система с правилами: AgNLP → Правила? → AgSrch/AgML	180	0,94
3	Система с 52 правилами: после 1000 обработанных писем	145	0,95
4	Система с 151 правилом: после 5000 обработанных писем	110	0,94
5	Система с 308 правилами: после 10000 обработанных писем	95	0,94

Конфигурация 1: Каждый документ проходит все этапы. Конфигурация 2: Система включает механизм проверки правил, но ни одно правило не срабатывает. Задержка в 180 мс показывает накладные расходы на саму проверку. Конфигурации 3, 4, 5 (рост базы правил): По мере того как AgRule анализирует историю решений, он выявляет частые шаблоны и создаёт соответствующие правила. Эти правила хранятся в метаграфе как RLME. При поступлении нового документа контейнерный агент AgAdapt сначала проверяет его соответствие существующим

щим правилам. Если находится совпадение, система пропускает медленные этапы AgSrch и AgML и сразу присваивает категорию. Это и даёт ускорение в 2.63 раз.

Кроме того, система способна на объяснение принимаемых решений, так как на каждом этапе работы взаимодействует с одной и той же информационной структурой. Например, для вакансии Тестировщик был рекомендован дополнительный навык «Методология тестирования», хотя в описании вакансии его не было (рисунок 8). При запросе на объяснение решения система отслеживает связи и возвращает информацию, что этот навык связан с требуемым в вакансии «Анализ тестирования» с достоверностью 0,8445.

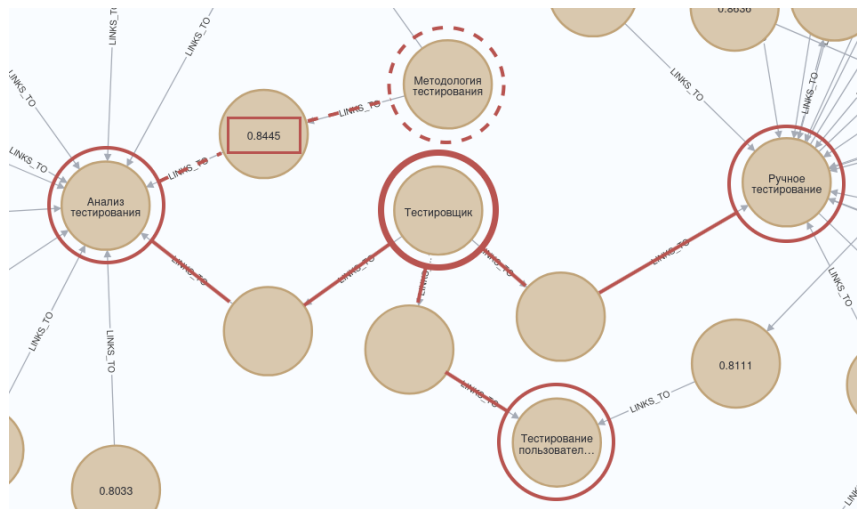


Рис. 8 – Фрагмент метаграфа, служащий для объяснения решений

Таким образом, СППР-ТДСП, построенная на информационной структуре метаграфа, обладает требуемыми свойствами: за счет наличия цепочки рассуждений в метаграфовой модели обеспечена объяснимость результата, модель знаний может самостоятельно расширяться в процессе работы, а последовательность обработки может изменяться на основе анализа среды. Реализованная архитектура гибридной СППР-ТДСП интегрирует фактографический анализ, семантическое расширение шинглов и гибридное ранжирование в единой метаграфовой модели, что соответствует второму пункту научной новизны. Кроме того, применение агентов, управляемых через метарёбра, позволило сократить количество фаз обработки за счёт раннего срабатывания сгенерированных правил – это реализует третий пункт научной новизны и демонстрирует повышение как адаптивности, так и объяснимости системы

ЗАКЛЮЧЕНИЕ

В ходе диссертационной работы получены следующие основные результаты:

1. Предложена и обоснована система утверждений информационного элемента метаграфа (ИЭМ), которая обеспечивает математическую основу для построения гибридных интеллектуальных информационных систем.

2. Разработана архитектура СППР, основанная на метаграфах для интеграции методов ML и символьных систем. Архитектура реализована в виде холонической многоагентной системы, где агенты организованы в иерархическую структуру. Центральным элементом архитектуры является контейнерный агент AgAdapt, который управляет набором специализированных агентов-функций (AgNLP, AgSem, AgSrch, AgML).

3. Реализован механизм самоорганизации, позволяющий системе адаптироваться без обучения с нуля. Механизм основан на анализе истории решений и генерации новых правил классификации, что делает систему способной к эволюции и адаптации к изменениям в предметной области.

4. Проведён сравнительный анализ эффективности предложенной архитектуры с традиционными подходами к представлению знаний, такими как онтология. Результаты показали, что предложенная архитектура превосходит традиционные подходы по таким критериям, как адаптивность, объяснимость и масштабируемость.

5. Оценен вклад архитектуры в повышение объяснимости и адаптивности системы. Объяснимость обеспечивается за счёт возможности проследить каждое решение через пути в метаграфе. Адаптивность достигается за счёт механизма самоорганизации, который позволяет системе эволюционировать на основе анализа данных.

6. Разработаны и программно реализованы методы работы с метаграфовой моделью, обеспечивающие согласованность и непротиворечивость знаний. Физическое хранение метаграфа в системе обеспечено с помощью графовой базы данных Neo4j, запросы к которой выполняются на языке Cypher. Логика метаграфовых агентов реализована с помощью языка Python 3 (свидетельства о государственной регистрации программы для ЭВМ №2025689047, №2025692521).

7. Проведена экспериментальная проверка и апробация разработанных методов и программных средств. Результаты тестирования показали, что система эффективно справляется с задачей классификации текстовых сообщений и может быть адаптирована для других предметных областей.

8. Предложена модель ранжирования, основанная на двухэтапном подходе: быстрый поиск кандидатов с помощью MinHash+LSH и точное ранжирование с помощью BM25F. Автором впервые предложена и разработана интеграция обоих этапов в виде метарёбер, что позволяет объединить статистический и символьный уровни обработки в рамках единой метаграфовой модели знаний.

СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ АВТОРОМ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в изданиях, рекомендованных ВАК РФ.

1. Дурова М.А., Зейн А.Н. Многоагентная организация системы обработки входящих писем с использованием метаграфовой модели // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2025. № 3. С. 1 – 13. DOI: 10.18137/RNU. V9187.25.03.P.1
2. Дурова М.А., Нечаев, А.А., Зейн, А.Н., Орлова, М.А. Частотное представление как перспективный метод оценивания качества изображений // Вестник ВГТУ. 2024. №3. С. 15-23. DOI: 10.36622/1729-6501.2024.20.3.002.

Публикации в других изданиях

1. Дурова, М. А. Разработка программного обеспечения для моделирования теплоэнергетического оборудования и схем с использованием динамических элементов / М. А. Дурова, Д. О. Цаплин, Р. О. Ивлёв // Радиоэлектроника, электротехника и энергетика : тез. докл. XXX Междунар. науч.-техн. конф. студентов и аспирантов (Москва, 29 февр. – 2 марта 2024 г.). – Москва: ООО «Центр полиграфических услуг „РАДУГА“», 2024. – С. 266.
2. Разработка информационной системы для моделирования работы энерготехнологического оборудования и энерготехнологических схем / Кан Д., Цаплин Д. О., Ивлёв Р. О., Дурова, М. А. // Технологии будущего: сборник трудов VII международной научно-технической конференции студентов и аспирантов, Москва, 22–26 июня 2023 года. – Москва: МЭИ, 2023. – С. 60-68.
3. Дурова, М. А. Разработка модели прогнозирования энергопотребления и цен в активных энергетических комплексах / М. А. Дурова, К. Мякота, Д. А. Фрей // Технологии будущего : VI Международная научно-техническая конференция студентов и аспирантов, Москва, 23–27 мая 2022 года / Национальный исследовательский университет «МЭИ». – Москва: Издательский дом МЭИ, 2022. – С. 114-119.
4. M. A. Durova, A. N. Zein, D. O. Tsaplin, D. S. Filippov, A. A. Krasnova and A. D. Ismailova, "Using Machine Learning Methods to Analyze Optimal Oil Drilling Sites," 2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2024, pp. 1-5
5. M. A. Durova, A. N. Zein, S. V. Borisova, A. A. Mishin, D. Kan and S. K. Osipov, "Autoregressive Models for Solving the Problem of Forecasting Active Energy Complexes," 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2023, pp. 1-6
6. M. A. Durova, E. A. Khodunov, A. S. Anikeeva, A. A. Mishin, A. N. Zein and S. V. Borisova, "Application of the dbscan Method to Solve the Problem of New Employees Adaptation," 2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2022, pp. 1-6
7. Комаров И.И., Осипов С.К., Зейн А.Н. Дурова М.А, Брызгунов П.А. Программное приложение для внесения, редактирования и обработки настроек программных комплексов для численного моделирования энергетических процессов, версия 2 // Свидетельство о государственной регистрации программы для ЭВМ № 2023686751 от 08.12.2023.
8. Дурова М.А Программное приложение для автоматической маршрутизации входящей корреспонденции на основе гибридной метаграфовой модели // Свидетельство о государственной регистрации программы для ЭВМ № 2025689047 от 24.10.2025.
9. Дурова М.А., Зейн А.Н., Орлова М.А., Борисова С.В. Программное приложение для аналитики рынка труда и рекомендации профессиональных навыков на основе метаграфовой модели // Свидетельство о государственной регистрации программы для ЭВМ № 2025692521 от 21.11.2025.