

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи



ГЛУХИХ ДМИТРИЙ ИГОРЕВИЧ

**МОДЕЛИ И АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ
ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ЭКСПЛУАТАЦИИ
СЛОЖНЫХ ТЕХНОЛОГИЧЕСКИХ ОБЪЕКТОВ**

Специальность: 1.2.2. Математическое моделирование, численные методы
и комплексы программ

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:

Кандидат физико-математических наук,
профессор

Захарова Ирина Гелиевна

Тюмень – 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1. АНАЛИЗ НАУЧНЫХ РАБОТ, ПОДХОДОВ И ПРАКТИЧЕСКИХ РАЗРАБОТОК.....	13
1.1 Принятие решения на основе мониторинга опасных ситуаций в задачах управления технологическими объектами	13
1.2 Системы поддержки принятия решений в задачах управления технологическими объектами	16
1.3 Рассуждение на основе прецедентов (CBR).....	25
1.4 Ключевые задачи реализации CBR-цикла в рамках ИСППР	27
1.5 Задача классификации состояний элементов сложного технологического объекта.....	30
Выводы по главе 1.....	36
ГЛАВА 2. МАТЕМАТИЧЕСКОЕ И ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ.....	38
2.1 Концептуальное моделирование сложного технологического объекта городской инфраструктуры	38
2.2 Представление сложного объекта и ситуаций на сложном объекте	42
2.3 Модели представления прецедентов в ситуационной базе знаний .	46
2.4 Моделирование процесса выбора решения.....	49
Выводы по главе 2.....	53
ГЛАВА 3. РЕШЕНИЕ ЗАДАЧ ОТБОРА СИТУАЦИЙ И АДАПТАЦИИ РЕШЕНИЙ	55
3.1 Обучаемая функция выбора решения	55
Выводы по разделу 3.1	62
3.2 Алгоритм адаптации и синтеза решения	63
Выводы по разделу 3.2	68
ГЛАВА 4. КОМПЛЕКС ПРОГРАММ	69
4.1 Логическая архитектура Платформы.....	69
4.2 Физическая архитектура Платформы	71
4.4 Алгоритм сборки предметно-ориентированной системы.....	74

4.5 Базовый набор модулей.....	74
4.6 Алгоритм генерации ситуаций (обучающих множеств).....	80
4.7 Разработка тестового набора прецедентов для ситуационной базы знаний.....	84
4.8 Задача оценки схожести ситуаций. Экспериментальная работа.....	93
Выводы по главе 4.....	102
ЗАКЛЮЧЕНИЕ	104
СПИСОК ЛИТЕРАТУРЫ	107
Приложение 1. Руководство пользователя.....	122
Приложение 2. Модуль оркестратор.....	149
Приложение 3. Свидетельства о регистрации РИД.....	155
Приложение 4. Акт внедрения.....	159

ВВЕДЕНИЕ

Актуальность темы исследования. Уровень цифровизации объектов современного производства и инфраструктуры позволяет переходить от задач сбора данных, прогнозирования событий и обнаружения нежелательных ситуаций к задачам интеллектуальной поддержки принятия своевременных решений в обнаруженных проблемных ситуациях.

Особенно актуально внедрение систем поддержки принятия решений (СППР) для сложных технологических объектов, где возникновение нештатных ситуаций несет за собой риски катастроф или иных серьезных экономических, экологических, социальных последствий. Примерами таких объектов являются системы обеспечивающих предприятий городской инфраструктуры (тепло-, водо-, газо-, энергоснабжение), объекты производства, объекты энергетики.

В условиях городской среды принятие решений для предотвращения нежелательных ситуаций и их последствий должно проводиться как с учетом состояния самого технологического объекта, так и с учетом его окружения, наличием ограничений и взаимосвязей с другими объектами города, а также состоянием обеспечивающих систем.

Таким образом, возникает комплексный, неоднородный и динамично меняющийся объект мониторинга (ОМ) с многообразием связей и состояний внутренней и внешней среды.

Согласно характеристикам сложного объекта, сформулированными Д.А. Поспеловым в методе ситуационного управления [1], такой ОМ относится к категории сложных, а его исследование требует привлечения методов математического моделирования и современных технологий анализа данных.

Многообразие окружения, динамичность состояний компонентов ОМ приводит к тому, что значительный объем данных необходимо анализировать, систематизировать и далее использовать для обеспечения принятия решений. В итоге, задачи внедрения инструментария прикладных интеллектуальных систем поддержки принятия решений имеют высокую трудоемкость и наукоемкость, что тормозит их решение в реальных условиях.

В силу этой трудоемкости, несмотря на широкое применение методов и средств прогнозной аналитики, мониторинга описанных сложных технологических объектов (СТО), остаются открытыми вопросы создания систем интеллектуальной поддержки принятия решений (ИСППР), которые имеют возможность генерировать обоснованные варианты разрешения нежелательных ситуаций и помогать при исполнении решений. В то же время возможности математического моделирования, а также современное состояние технологий, уровень цифровизации процессов, в том числе, систем сбора и аналитики данных, необходимых для реализации таких моделей, позволяют уже сегодня переходить к созданию таких ИСППР.

Для создания ИСППР в системах сложных технологических объектов необходима интеграция разных методов искусственного интеллекта, что дает возможность объединить преимущества систем на основе машинного обучения, так и систем, основанных на знаниях. Это позволит преодолеть такие трудности, как отсутствие необходимых объемов обучающих данных, слабая формализуемость и сложность формирования строгих математических критериев и целевых функций для принятия решений на сложных объектах. Соответственно, необходим гибридный подход.

Метод рассуждения на основе прецедентов (ситуационный подход, case-based reasoning, CBR) рассматривается в качестве базы гибридного подхода. В рамках подхода предлагаемый гибридный метод потенциально позволяет устранить ограничения раздельного применения систем, основанных на знаниях (требуют трудоемкой работы по выявлению и формализации знаний) и методов машинного обучения (требуют большие объемы обучающих примеров).

Степень разработанности темы. Исследованиям в области математического моделирования, формализации состояния технологических объектов посвящены работы следующих авторов: А.Ф. Антипин, В. К. Битюков, В. С. Кудряшов, М. В. Алексеев, Д. П. Лащенко, В. Л. Бурковский, Т. Н. Караневская, А. Г. Шумихин. В работах используются подходы к моделированию объекта с помощью уравнений, структурных моделей. Исследования в области

разработки СППР для технологических объектов посвящены работы следующих ученых: Поспелов Д.А., Охтилев М.Ю., Спирин Н. А., Онорин О.П., Лавров В.В., Соколов Б.В., Юсупов Р.М. Исследователи предлагают концепции СППР на основе знаний, математических моделей для прогноцирования состояний. Исследованиям в области применения метода рассуждения на основе прецедентов для интеллектуальных СППР посвящены работы следующих исследователей: Карпов Л.Е., Юдин В.Н., Еремеев А.П., Варшавский П.Р., Юрин А.Ю., Кузяков О.Н., Aamodt A., Pinar Öztürk, Ramon López de Mántaras, Rosina Weber, Ian Watson. В рамках реализации цикла CBR можно отметить работы, связанные с интегративным подходом на основе разных моделей, следующих авторов: Грибова В.В., Eoin M. Kenny, Bjorn Magnus Mathisen, Maximilian Hoffmann, Agnar Aamodt и др. Для управления технологическими объектами известны системы: IBM Watson, SAS Decision Manager, AnyLogic, а также отечественные системы корпораций Газпромнефть, Роснефть, Алроса, Северсталь и другие.

Однако остаются открытыми вопросы математического моделирования и алгоритмического обеспечения ИССПР для сложных технологических объектов, где необходимо учитывать мультимодальные данные от компонентов, контекст (условия в которых эксплуатируется объект в момент принятия решения), а также брать во внимание возможность недостатка обучающих данных, неопределенность состояний.

В контексте актуальной проблемы необходимого повышения оперативности и адекватности принятия решения по корректировке/восстановлению работоспособности в процессе эксплуатации СТО на основе анализа разработанности темы в исследовании поставлена научная задача.

Научная задача, решаемая в диссертации, заключается в разработке методов и алгоритмов для задач интеллектуальных систем поддержки принятия решений, а также разработке комплекса программ – Платформы для быстрой сборки интеллектуальных систем поддержки принятия решений, реализующего предложенные методы и алгоритмы. Ее решение имеет научную и практическую ценность для построения интеллектуальных систем поддержки принятия решений

при эксплуатации сложных технологических объектов, для которых значимы оценка комплексной ситуации на объекте с учетом контекста и быстрая выдача решения.

Объект исследования: математическое моделирование ситуации на сложном технологическом объекте.

Предмет исследования: методы моделирования состояний сложных технологических объектов в интеллектуальных системах поддержки принятия решений; метод рассуждения на основе прецедентов в задачах поддержки принятия решений; нейросетевой подход для выбора решения; алгоритмы генерации обучающих множеств; алгоритмы адаптации решений.

Цель диссертационной работы: разработка, исследование и программная реализация моделей и алгоритмов для обеспечения быстрой выдачи, интерпретируемости и аргументированности рекомендаций в системах поддержки принятия решений на сложных технологических объектах.

Для достижения цели необходимо решить следующие задачи:

1. Разработать метод математического моделирования ситуации на сложном технологическом объекте;
2. Разработать и исследовать методы моделирования процесса принятия решения и алгоритмы построения и выдачи решения, соотнесенного с конкретным набором состояний компонентов сложного технологического объекта;
3. Реализовать разработанные методы и алгоритмы в программном комплексе: Платформа для быстрой сборки интеллектуальных систем поддержки принятия решений при эксплуатации сложных технологических объектов;
4. Провести апробацию Платформы на примере тестовых сборок для сложных технологических объектов: тепловой пункт, умная ферма, понижающая электроподстанция.

Научная новизна:

1. Разработан метод математического моделирования ситуаций на сложных технологических объектах, *отличающийся* новым способом представления

ситуации через состояния компонентов объекта и его контекста в виде векторов состояний, *что позволяет* единообразно представить различную информацию, поступающую от компонентов, для моделирования комплексной ситуации на объекте с учетом контекста и осуществлять отбор прецедента из базы знаний в едином пространстве состояний с инвариантной относительно компонентов метрикой. (п. 6)

2. Разработан численный метод моделирования построения решения на базе гибридного метода рассуждения на основе прецедентов, *отличающийся* совместным использованием систем, основанных на знаниях, и методов машинного обучения, *что позволяет* устранить ограничения отдельного применения этих подходов: системы, основанные на знаниях, требуют трудоемкой работы по выявлению и формализации знаний; методы машинного обучения требуют большие объемы обучающих примеров. (п. 2)

3. Разработан алгоритм сборки и адаптации решения, *отличающийся* применением разработанного метода моделирования состояний сложных технологических объектов. Алгоритм *позволяет* в случае отсутствия в базе знаний подходящего решения на основе имеющихся прецедентов синтезировать новое решение путем сборки решения из имеющихся в базе элементарных преобразований состояний компонентов объекта и последовательностей таких преобразований. (п. 8)

4. Разработан комплекс программ, реализованный в виде Платформы для быстрой сборки интеллектуальных систем поддержки принятия решений, отличающийся микроядерной архитектурой, в частности, составом модулей и их взаимодействием, *что позволяет* собирать и настраивать из готовых модулей предметно-ориентированную ИСППР и осуществлять поддержку принятия решений в ходе эксплуатации на основе гибридного метода рассуждения на основе прецедентов (п. 3).

Теоретическая значимость диссертационного исследования заключается в разработке: понятие *сложного технологического объекта*, объединяющего в себе predetermined связи и состояния не только технологических компонентов, но

и персонала, зданий и сооружений, подъездных путей и окружающей среды; понятие *ситуации* на сложном технологическом объекте, подразумевающую набор состояний компонентов объекта; метод моделирования состояний сложных технологических объектов, позволяющего единообразно формализовать различную информацию от разнородных элементов; метод гибридного рассуждения на основе прецедентов, позволяющего устранить ограничения раздельного применения систем, основанных на знаниях (требуют трудоемкой работы по выявлению и формализации знаний) и методов машинного обучения (требуют большие объемы обучающих примеров).

Практическая значимость диссертации заключается в разработке комплекса программ – Платформы для быстрой сборки интеллектуальных систем поддержки принятия решений, реализующего предложенные методы и алгоритмы и позволяющего собирать и настраивать из готовых модулей предметно-ориентированную ИСППР для технологических объектов, для которых значимы оценка комплексной ситуации на объекте с учетом контекста и быстрая выдача решения.

Методы исследования. Использованы методы инженерии знаний, математического моделирования, машинного и глубокого обучения, метод рассуждения на основе прецедентов, методы оптимизации, методы объектно-ориентированного программирования, методы проектирования программных комплексов (микроядерная архитектура).

Положения, выносимые на защиту

1. Метод математического моделирования ситуации на сложном технологическом объекте, которая описывается совокупностью состояний компонентов объекта и его контекста в виде векторов состояний;

2. Численный метод моделирования процесса построения решения при эксплуатации сложного технологического объекта на базе гибридного метода рассуждения на основе прецедентов, включая обучаемую функцию выбора решения;

3. Алгоритм сборки и адаптации решения на основе имеющихся прецедентов в базе знаний;

4. Платформа для быстрой сборки интеллектуальных систем поддержки принятия решений, позволяющая собирать и настраивать из готовых модулей предметно-ориентированную ИСППР, функционирующую на основе разработанных методов и алгоритмов, включая алгоритм генерации обучающих множеств.

Соответствие диссертации паспорту научной специальности.

П2. Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий: для эффективного построения решения в ИСППР на сложном технологическом объекте разработан, обоснован и протестирован численный метод моделирования процесса построения решения на базе гибридного метода рассуждения на основе прецедентов, включая алгоритм сборки и адаптации решения на основе имеющихся прецедентов в базе знаний. Метод реализуется с применением современных компьютерных технологий (искусственные нейронные сети, в том числе, на основе предобученных моделей компьютерного зрения).

П3. Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента: предложенные методы и алгоритмы реализованы в виде комплекса программ, на базе которого проведены вычислительные эксперименты по оценке эффективности разработанного алгоритмического обеспечения и возможности его практического использования для решения поставленных задач.

П6. Разработка систем компьютерного и имитационного моделирования, алгоритмов и методов имитационного моделирования на основе анализа математических моделей: для анализа ситуации на сложном технологическом объекте разработан метод математического моделирования ситуации на таком объекте. Анализ соответствующей математической модели ситуации на объекте

позволяет реализовать компьютерное моделирование процесса построения решения.

П8. Комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента: в рамках исследования научно-технической проблемы разработки интеллектуальных систем поддержки принятия решений для технологических объектов разработан современный метод математического моделирования ситуации на сложном технологическом объекте, включающий в себя совместное использование знаний и методов машинного обучения. Разработаны алгоритмы сборки и адаптации решения, генерации обучающих множеств, решающие актуальные научно-технические проблемы недостатка знаний. Проведены вычислительные эксперименты.

По теме диссертационного исследования опубликовано 18 работ, в том числе: 8 в рецензируемых журналах из перечня ВАК РФ; 4 в научных журналах, индексируемых в Scopus, 3 в материалах конференций, индексируемых в Scopus, 3 в материалах конференций, индексируемых в РИНЦ. Получено 3 свидетельства о регистрации программ для ЭВМ.

Степень достоверности результатов. Результаты работы согласуются с результатами известных работ по соответствующей тематике, опубликованных другими отечественными и зарубежными исследователями. Используются адекватные поставленным задачам метрики оценки качества моделей машинного обучения. При оценке результатов в ходе апробации привлечены эксперты.

Апробация результатов работы.

Основные результаты, полученные в настоящей диссертационной работе, были представлены в виде докладов на следующих конференциях:

- Международная научно - практическая конференция «Информационные технологии и интеллектуальные системы принятия решений» (ITIDMS 2021). Место проведения РосНОУ, ул. Радио, 22, Москва (2021)

- X Научно - практическая конференция БРД "Газпром нефть". 2021. Место проведения: ООО Газпромнефть Научно - Технический Центр, Набережная реки Мойки. д.75 - 79, лит Д, г. Санкт - Петербург (2021).
- 2nd International Conference on Data Science and Applications (ICDSA 2021). Место проведения: School of mobile computing and communication, Jadavpur university, Kolkata, India (2021)
- Математическое и информационное моделирование. Место проведения: ТюмГУ, Тюмень (2023)
- Новые информационные технологии в нефтегазовой отрасли и 15 мероприятия образования Место проведения: Институт геологии и нефтегазодобычи, ТИУ, Тюмень (2022, 2024)
- International Conference on Intelligent Information Technologies for Industry (ИТИ). Место проведения: Харбин, Китай (2024).

Структура и объем диссертации. Диссертация изложена на 160 страницах машинописного текста, содержит 23 рисунков, 4 таблицы, состоит из введения, четырех глав, заключения, списка использованной литературы из 121 наименований и 4 приложений на 38 страницах.

Сведения о личном вкладе автора. Формулировка народно-хозяйственной проблемы, постановка научной задачи, содержание диссертации и все представленные в ней результаты получены лично автором. Подготовка к публикации некоторых научных статей, а также получение и регистрация результатов интеллектуальной деятельности проводились совместно с соавторами, вклад диссертанта определяющий.

ГЛАВА 1. АНАЛИЗ НАУЧНЫХ РАБОТ, ПОДХОДОВ И ПРАКТИЧЕСКИХ РАЗРАБОТОК

1.1 Принятие решения на основе мониторинга опасных ситуаций в задачах управления технологическими объектами

Исследование касается информационных процессов мониторинга для принятия решений в задачах управления технологическими объектами (ТО). Такие объекты могут возникать в системах городской инфраструктуры (объекты тепло-, энерго-, водо-, газоснабжения), на производствах, в добывающей промышленности.

Управление сложными технологическими объектами включает в себя множество задач, связанных с проектированием, контролем и оптимизацией процессов. В качестве ключевых направлений исследований и прикладных разработок в этой области можно выделить следующие:

1. Разработка методов управления. Подразумевает, в том числе, путем создания инструментальных программных средств, ориентированных на решение задач автоматизации процессов и повышении эффективности управления.

2. Проектирование систем управления. Включает в себя определение целей, критериев эффективности и ограничений, что позволяет создать концептуальные модели для управления сложными объектами и процессами.

3. Автоматизация системы управления. Этот процесс позволяет анализировать недостатки существующих систем и предлагать модели для их улучшения. Например, формальное описание системы управления на начальном этапе автоматизации может быть использовано для тестирования алгоритмов управления.

4. Повышение безопасности и эффективности. Основные цели автоматизации технологических процессов заключаются в повышении безопасности и эффективности производственного процесса. Это достигается через внедрение современных технологий и методов управления.

5. Внедрение управления на основе данных, в том числе с применением нейροкомпьютерных технологий. Современные подходы к управлению также включают использование нейрокомпьютерных технологий, которые повышают эффективность обработки данных и контроля за технологическими объектами.

6. Многокритериальный подход к управлению. Управление сложными системами часто требует многокритериального подхода, где в качестве основных инструментов может выступать имитационное моделирование для анализа и оптимизации процессов. Это позволяет более точно оценивать влияние различных факторов на систему.

Научными работами в данной области занимались отечественные и зарубежные исследователи: Башлыков А.А., Кузяков О.Н., Мышляев Ю.И., Лавров В.В., Гатилов М.А., Ковалёв С.П., Королёва М.Н., Хадеев А.С., Sayed K, Jiang Z, Kiziroglou M.E., Barriquello. C, Rukmani P. и другие.

Функционирование технологических объектов (ТО) сопровождается рисками возникновения аварийных ситуаций и негативных последствий, которые могут быть ими вызваны. При этом в условиях городской среды принятие решений для предотвращения нежелательных ситуаций и их последствий должно проводиться как с учетом состояния самого ТО, так и с учетом его окружения, наличием ограничений и взаимосвязей с другими объектами города, а также состоянием обеспечивающих систем.

Таким образом, возникает комплексный, неоднородный и динамично меняющийся объект мониторинга (ОМ) с многообразием связей и состояний внутренней и внешней среды.

Согласно характеристикам сложного объекта, сформулированными Д.А. Поспеловым в методе ситуационного управления [1], такой ОМ относится к категории сложных, а его исследование требует привлечения методов математического моделирования и современных технологий анализа данных.

Введем определение: сложный технологический объект – комплексный, неоднородный и динамично меняющийся объект мониторинга со множеством

предопределенных связей и состояний технологических компонентов и окружающей среды.

Эксплуатация таких объектов сопровождается задачами предотвращения нештатных, опасных ситуаций [2-5], задачами нейтрализацией последствий возникших аварийных ситуаций. Несвоевременное принятие верного решения приводит к нежелательным, в том числе, катастрофическим последствиям.

В то же время, принятие решений, связанных с предотвращением опасных ситуаций и нейтрализации их последствий, является высоко ответственной задачей. Для таких ситуаций характерны: дефицит времени на принятие решений, большое количество участников, недостаток знаний о всех условиях произошедшей ситуации. Высоки риски принятия малоэффективного, неверного решения или вовсе его отсутствия. Комплексная поддержка лица принимающего решения является актуальной.

Однако на данный момент системы поддержки работоспособности заключаются в удаленном контроле технологических объектов и мониторинге их параметров. Подобные распространенные умные производства, умные объекты инфраструктуры имеют в основе инструменты мониторинга и анализа. В том числе, беспилотные летательные аппараты, искусственный интеллект на основе предиктивной аналитики.

Например, развертывание современной городской энергосети подразумевает объединение систем SCADA с системами передачи данных [6], внедрение цифрового двойника [7,8], который отразит полный цикл работы системы и покажет сроки износа оборудования в зависимости от условий эксплуатации.

Отдельное направление научных исследований рассматриваемой области составляют работы, посвященные техническому и технологическому аспекту проблемы [9-13]. Известны отечественные и зарубежные исследователи Сысоев Ю.С., Спирин Н.А., Сальников А.А., Лавров В.В., Бекетов В.Г., Чернов А.В., Шпицер В.Я., Кривин В.В., Толстов В.А., He, P., Wang, J., Chelaru E, Grigoras G. и другие.

В работах [14-17] предлагается применять методы интеллектуального анализа данных и (или) нейросетей для выявления нештатных ситуаций. Ряд исследований касается методологической поддержки принятия решений для планирования обслуживания элементов [18-19], известны предложения о применении методов интеллектуального анализа данных и (или) нейросетей для прогнозирования спроса ресурсов распределяющей системы [20], анализ производственных ситуаций на основе моделирования процесса [21], прогнозирования неисправности отдельных элементов [22], планированию обслуживания [23].

Упомянутые выше исследования опираются на подготовку и использования значительного объема обучающих данных (примеров ситуаций). Отметим, что в случае сложного объекта мониторинга обучающих данных в необходимом количестве может и не быть.

Другое направление работ связано с экспертно-советующими системами, в основе которых лежит инженерия знаний [24-26]. Проведенный анализ исследований показал, что данные системы сопровождаются возникновением проблемы трудоемкости выявления и формализации знаний, которая ограничивает их использование относительно простыми объектами и ситуациями.

Суммируя вышесказанное, можно говорить о направленности теоретических исследований и прикладных разработок в сторону проблем именно сбора в реальном времени первичных данных, которые позволяют делать вывод об актуальном состоянии объекта мониторинга и прогнозировать изменение его состояния и состояния отдельных элементов и параметров.

1.2 Системы поддержки принятия решений в задачах управления технологическими объектами

Отдельного рассмотрения требуют системы поддержки принятия решений (СППР). Они обеспечивают эффективное и обоснованное принятие решений в

условиях неопределенности и сложности, обусловленной множественными и разнородными данными.

СППР — это компьютерные системы, которые помогают пользователям анализировать данные и принимать решения на основе различных критериев и факторов. Они учитывают как объективные, так и субъективные аспекты, влияющие на процесс принятия решений, включая оценку рисков и альтернатив. Такие системы становятся особенно актуальными в условиях сложных технологических процессов, где требуется быстрая и точная реакция на изменения.

Современные методы построения СППР берут начало в работах отечественных и зарубежных исследователей: Поспелова Д. А., Виттих В. А., Даса С., Джоханса Р., Финна В. К., Голицына Г. А., Хаагзмы И., Осипова Г. С., Осипова В. П., Ларичева О. И., Турбана Э., Фокса Д. [1, 27-32].

Развитие систем поддержки принятия решений складывается на основе знаний и моделей. Вклад в развитие внесли отечественные ученые: Поспелов Д.А., Охтилев М.Ю., Спирин Н. А., Онорин О.П., Лавров В.В., Соколов Б.В., Юсупов Р.М. и др. И зарубежные ученые: Грубер Т.Р., Гуарино Н., Ниренбург С., Грозоф В., Штааб С., Саати Т.

Так в [33, 34] представлено построение систем на основе инженерии знаний, онтологии, графов знаний. В работах [35-37] рассматриваются подходы к проектированию систем поддержки принятия решений для управления технологическими процессами в металлургии. В работах [38, 39] представлены исследования разработки СППР на основе теории многокритериального оценивания.

Существует ряд принципиально отличающихся архитектур СППР, каждая из которых имеет свои особенности и области применения. Наиболее распространенные архитектуры включают:

- Традиционные системы: Основываются на анализе данных и предоставляют пользователю инструменты для обработки информации.

- Интеллектуальные системы: Используют алгоритмы машинного обучения и искусственного интеллекта для улучшения качества принимаемых решений.
- Интегрированные системы: Объединяют различные источники данных и обеспечивают комплексный подход к анализу и принятию решений.

Современные СППР используют различные технологии для автоматизации процессов принятия решений. В том числе:

- Аналитические системы: Позволяют обрабатывать большие объемы данных и выявлять закономерности, что способствует более обоснованным решениям.
- Хранилища данных: Обеспечивают централизованное хранение информации, что упрощает доступ к данным и их анализ.
- Методы оптимизации: Используются для нахождения наилучших решений в условиях ограничений и неопределенности.

Использование систем поддержки принятия решений в управлении технологическими объектами позволяет достигнуть: скорости принятия решений, т.к. автоматизация процессов позволяет быстрее реагировать на изменения в производственной среде, потенциально повысить эффективность решений: доступ к аналитическим инструментам и данным повышает обоснованность принимаемых решений. Также внедрение СППР позволяет потенциально снизить затраты эксплуатации.

СППР находят широкое применение в различных областях управления технологическими объектами, включая:

- Производственные процессы: Оптимизация производственных потоков и управление ресурсами.
- Управление рисками: Оценка и минимизация потенциальных рисков, связанных с технологическими процессами.

- Прогнозирование: Использование исторических данных для предсказания будущих событий и тенденций.

Существующие программно-технические решения, связанные с мониторингом и принятия решений на основе баз знаний являются или узкоспециализированными, или недостаточно направленными, другими словами такие системы недостаточно функциональные и малоподходящие для решения задач поддержки принятия решений на сложном технологическом объекте.

Далее представлены существующие платформенные решения потенциально применимые для разработки СППР на технологических объектах:

«Диспетчер» — система класса MDC (Machine Data Collection), которая собирает данные о работе всех производственных объектов (оборудования, рабочих мест основных сотрудников, сервисных служб и т. д.) для управления производством. Основное предназначение системы: мониторинг, анализ данных; функционал поддержки принятия решений в виде вывода вариантов решений не предусмотрен.

zenon — программный пакет SCADA для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, АСКУЭ. Основное предназначение системы: мониторинг, функционал ИСППР в виде вывода вариантов решений не предусмотрен.

Maana – продукт для разработки приложений, повышающих операционную эффективность и производительность. В отличие от предлагаемого проекта Maana решает более глобальные задачи и нацелена на упрощение именно разработки приложений, позволяющих организовать разнородные корпоративные знания, аналитические инструменты и обеспечить к ним доступ на единой платформе . Возможности построения ситуационных баз знаний с автоматическим выводом решений в критических ситуациях не предусмотрены.

Loginom — low-code платформа, предназначенная для формирования систем сбора и аналитики больших данных. Функционал построения ситуационных баз знаний и вывода вариантов решений в критических ситуациях не предусмотрен.

jCOLIBRI — объектно-ориентированный фреймворк на Java для построения систем вывода, основанных на прецедентах (CBR). Основной функционал нацелен на упрощение программирования и отладку приложений, использующих CBR. Создаваемые решения применимы для разных областей, работают в диалоговом режиме с пользователем для уточнения запросов в ходе прецедентного вывода. Однако, реализованы наиболее общие и простые способы отбора прецедентов, основанные на простых метриках и экспертных правилах. Не предусмотрена возможность использования методов машинного обучения и нейросетей.

На зарубежном рынке известны следующие системы поддержки принятия решений:

1. IBM Watson

Эта система использует технологии искусственного интеллекта для анализа больших данных и поддержки принятия решений в различных отраслях, включая производственные объекты.

2. SAP BusinessObjects

Это мощная платформа для бизнес-аналитики, которая предоставляет инструменты для анализа данных и визуализации, что помогает в принятии обоснованных решений на уровне управления.

3. Microsoft Power BI

Эта система позволяет пользователям собирать, анализировать и визуализировать данные, что способствует более эффективному принятию решений на основе аналитики.

4. Oracle Crystal Ball

Это программное обеспечение для прогнозирования и анализа рисков, которое помогает пользователям моделировать различные сценарии и оценивать потенциальные результаты.

5. Tableau

Платформа для визуализации данных, которая позволяет пользователям создавать интерактивные отчеты и дашборды, что облегчает процесс принятия решений на основе данных.

6. QlikView

Система предоставляет инструменты для анализа данных и создания отчетов, что помогает в принятии решений на основе глубокого анализа информации.

7. SAS Decision Manager

Система поддерживает автоматизацию процессов принятия решений и позволяет интегрировать аналитические модели в бизнес-процессы.

8. AnyLogic

Моделирующая платформа, которая используется для создания имитационных моделей и анализа сложных систем, что помогает в принятии решений в области управления.

9. Palisade @RISK

Программное обеспечение для анализа рисков, которое использует метод Монте-Карло для оценки неопределенности и поддержки принятия решений в сложных ситуациях.

Также существует ряд некоммерциализируемых систем поддержки принятия решений, разрабатываемых крупными промышленными корпорациями (Газпромнефть, Роснефть, Алроса, ММК, Северсталь и т.п.). Их системы специализированы под задачи внедряемых производств и часто узконаправленны.

В отношении технологических объектов задачи предупреждения и устранения опасных ситуаций связаны одновременно как с задачами мониторинга параметров объекта, так и с задачами оперативной поддержки решений, направленных на нейтрализацию последствий нежелательных ситуаций.

В российских компаниях можно выделить следующие примеры:

1. Газпром нефть

Компания Газпром нефть является одним из наиболее успешных примеров применения систем поддержки принятия решений в России. Она использует СППР для оптимизации производственных процессов на своих месторождениях и повышения эффективности добычи. Разработанная с учетом специфических потребностей компании, система автоматически собирает и анализирует данные, предоставляя информацию о процессах добычи в режиме реального времени.

2. Северсталь

Компания Северсталь использует СППР для оптимизации своих производственных процессов. Система анализирует данные о производственных процессах на заводах и помогает определить оптимальную схему работы, учитывая такие факторы, как затраты на энергию, материалы и трудовые ресурсы.

3. Роснефть

«Цифровое месторождение» — это система, модуль СППР которой принимает сигналы с множества датчиков, установленных на механизированном фонде скважин месторождения. Она выполняет диагностику состояния оборудования, прогнозирует возможные события и предоставляет рекомендации. Другие модули системы связаны с промышленной безопасностью, охраной труда, а также с функционированием насосов, трубопроводов и другими процессами.

Однако на фоне активного внедрения систем сбора и анализа данных, задача оперативной поддержки принятия решений в полной мере не решается. Реализация в одном программно-техническом комплексе функционала, направленного на решение двух выделенных выше задач совместно (анализ ситуации и предложение решения), определяет современную концепцию систем поддержки принятия решений [4, 5] – интеллектуальных ИСППР.

Интеллектуальные СППР (ИСППР) должны обладать возможностями не только прогнозировать и выявлять нежелательные ситуации, но и генерировать варианты разрешения ситуаций в реальном времени, а также обеспечивать поддержку исполнения решений [2,5,40].

Необходимость исследований путей и подходов к реализации современной концепции ИСППР обуславливается не только актуальностью поддержки решений, но и возможностями практического применения и внедрения такой ИСППР на реальный сложный технологический объект, что сегодня, в многом реализовано благодаря развитым методам и технологиям сбора и анализа данных. Таким образом, с одной стороны, актуальным является внедрение ИСППР в системы управления и эксплуатации сложных объектов; с другой стороны, современный уровень цифровизации процессов, технологий и методов сбора, анализа данных обуславливает технологические возможности для практической реализации таких ИСППР. В то же время практика внедрения ИСППР ограничивается еще трудоемкостью и наукоемкостью решаемых задач. Каждая разработка прикладных ИСППР требует привлечения как квалифицированных научных сотрудников – специалистов в данной области, так и инженерно-технический персонал для проведения программных разработок. Как следствие, становится особенно актуальным решение задач, направленных на создание типовых программных решений, фреймворков и платформ ИСППР, которые будут базироваться на специализированных научных методах, моделях и алгоритмах интеллектуальной поддержки принятия решений.

Исследования обозначенной проблемы сосредотачиваются на решении прикладных задач для разработки предметно-ориентированных ИСППР на основе известных подходов и на создании новых методов, которые позволяют объединить возможности разных инструментов искусственного интеллекта – машинного обучения и «классического» ИИ, основанного на знаниях. Это позволяет преодолеть недостатки и трудности их отдельного применения, чтобы получить результаты, обеспечивающие необходимое качество решений прикладных ИСППР в условиях неопределенности, при недостатке данных и знаний.

Исследованию вопросов построения интеллектуальных систем посвящены работы Поспелов Д.А., Пальчевский Е. В., Голицына Г. А., Фоминых И. Б., Поспелова Д. А., Комарцовой Л. Г., Шавранский В. Малтугова Г.С., Юрин А.,

Рахманкулова В. З., Финна В. К., Кузнецова О. П., Соттара Д., Тарека Р., д'Авилы Гарсеа А., Медскера Л., Хитцлера П., Гинзбурга М., Галланта С. и др.

На сегодняшний день наиболее проработано совместное использование разных моделей из арсенала «классического» ИИ. В этом направлении можно отметить ряд современных публикаций, где для вывода решений в условиях слабой формализации и высокой неопределенности применяется совместное использование:

- логических и расчетно-логических, аналитических моделей, включая модели группового выбора, многокритериальных решений, анализа иерархий и т.п. [41-43];
- онтологий и вывода на прецедентах [42, 44];
- продукционных правил и нечетких продукционных систем с выводом на прецедентах и онтологиями [45, 46];
- нейросетевая обработка больших данных [47].

Анализ научных результатов в данной области показывает, что остаются открытые вопросы. Более того, широкое внедрение в современную производственную деятельность систем поддержки принятия решений, опирающихся на методы искусственных нейросетей и машинного обучения, выявило проблемы, связанные с недостатками этих моделей (необходимость больших объемов обучающих данных, низкая объясняющая способность, отсутствие аргументации результатов и др.) [48].

Актуальность приобретает направление исследований, в рамках которого развивается совместное использование методов машинного обучения и методов систем, основанных на знаниях. Совместное использование потенциально позволит решить ряд вышеперечисленных проблем. Для реализации совместного использования необходима разработка и обоснование соответствующих моделей и алгоритмов.

1.3 Рассуждение на основе прецедентов (CBR)

Перспективным для совместного использования является ситуационный метод (вывод решения на основе прецедентов, case-based reasoning, CBR) [1, 49-55], в котором в зависимости от внешних условий и выбранных акцентов могут воплощаться как модели инженерии знаний, так и машинного обучения.

Применение ситуационного подхода в задачах управления и поддержки принятия решений на технологическом объекте исследуют отечественные ученые: Поспелов Д.А., Филимонов Н.Б., Еремеев А.П., Варшавский П.Р., Виттих В.А. и др. И зарубежные ученые: Gary M., Aamodt A., Ramon López de Mántaras, Huang, K., Jiang, X.

Ситуационный метод применён с положительным результатом во многих предметных областях. В том числе, в области принятия решения при эксплуатации сложных технических, технологических объектов [53, 54]. Применение данного метода для задач интерпретации данных, прогнозирования развития событий и вывода рекомендаций различным участникам процесса управления объекта мониторинга в комплексной постановке ранее не решались.

Идея CBR основана на применении решения из опыта: при возникновении новой ситуации, требующей действий, в базе знаний ИСППР находится схожий прецедент – схожая ситуация и ее решение. Решение из найденного прецедента используется в новой ситуации. База знаний пополняется в начале развертывания ИСППР и пополняется в ходе эксплуатации системы. Прецеденты представляют собой примеры ситуаций и их решений из опыта и ситуаций, смоделированных экспертами.

Таким образом, реализуется процесс, названный CBR-циклом [55], который включает в себя этапы:

1. идентификация актуальной ситуации на объекте мониторинга, поиск в БЗ и выбор ситуации, наиболее схожей с актуальной;
2. выбор из пары ситуация-решение решения для его повторного использования;

3. оценка и, при необходимости, адаптация извлеченного из пары решения в новое решение;
4. передача выбранного решения на исполнение;
5. оценка результата и сохранение новой пары ситуация-решение в БЗ.

Метод рассуждения на основе прецедентов обладает рядом положительных качеств [2,5,40,56,57]:

- скорость выдачи решения – решение не разрабатывается с нуля, а используется готовое;
- сбор прецедентов – база знаний пополняется в процессе эксплуатации;
- возможность дообучения за счет адаптации решений;
- логичный и простой способ представления для пользователей знаний;
- возможность соединить два ключевых подхода к построению систем поддержки решений – системы на знаниях и традиционное машинное обучение.

В рамках реализации цикла CBR можно отметить работы, связанные с интегративным подходом разных моделей следующих авторов: Грибова В.В., Eoin M. Kenny, Bjorn Magnus Mathisen, Maximilian Hoffmann, Agnar Aamodt и др.:

- применение машинного обучения, в том числе обучения с подкреплением, нейросетей для оценки схожести, формирования и настройки метрик для функции определения схожести при отборе прецедентов в базах знаний [58-62];
- адаптация прецедентов для пополнения БЗ с помощью машинного обучения и нейронных сетей [63-66];
- выявление в больших объемах данных атрибутов для формирования базы прецедентов [67];
- применение CBR для задач генерации объяснений результатов моделей машинного обучения [68];

– применение методов инженерии знаний, онтологий для представления прецедентов и извлечения прецедентов с использованием знаний [69-72].

Примером работы, где рассматривается весь цикл - «Обучение (формирование знаний) - Вывод решений (применение) – Дообучение (расширение знаний)» является исследование [73]. Здесь предложена архитектура, позволяющая решать комплекс задач: распознавания ситуаций с помощью нейросетей – вывод решений с помощью CBR – формирование прецедентов.

Задачи формирования прецедентных баз знаний, оценки их полноты, актуализации и обслуживания базы прецедентов могли бы рассматриваться как самостоятельное направление исследований в области рассматриваемой проблемы. Однако, исследователи решают эти задачи в рамках общего комплекса задач разработки прикладных ИСППР, уделяя больше внимания представлению прецедентов и поиску в базе знаний.

1.4 Ключевые задачи реализации CBR-цикла в рамках ИСППР

Для реализации CBR-вывода в каждой конкретной предметной области необходимо решить множество задач. Одной из ключевых является задача идентификации и выбора схожих ситуаций. Принятие решений в условиях городской инфраструктуры требует учета множества факторов.

Это не только текущее состояние технологического оборудования, но и состояния операционного окружения, погодные условия, доступность ресурсов и другое. Большое количество факторов принятия решений влечет за собой возрастание количества возможных ситуаций. Это усложняет их формализованное описание и сравнение для принятия решений.

Классические методы сравнения, основанные на метриках расстояний в пространстве параметров, часто оказываются недостаточно точными в подобных условиях. Поэтому современное развитие CBR-систем направлено на поиск новых подходов к сравнению и выбору ситуаций в сложных предметных областях. [74-77].

Разработка и исследование метода оценки схожести остаются актуальными на протяжении многих лет [78-80]. Идея интеграции CBR и нейросетей также рассматривалась ранее. Возможности применения нейронных сетей для оценки близости ситуаций активно изучаются. Исследования последних лет показывают положительные результаты.

В исследовании [81] предложена типология моделей для оценки функций схожести, которая включает два ключевых этапа: выделение значимых признаков и формирование эмбедингов, а также сравнение этих эмбедингов для оценки сходства ситуаций. Важно учитывать, какая информация используется для описания атрибутов сложной ситуации, а также сколько времени и усилий потребуется для формализации исходных данных в условиях возникновения нежелательной ситуации.

В работе [75] описан эксперимент, в котором оценка качества автомобиля осуществляется через его схожесть с другими известными автомобилями в пространстве параметров. Для этого применяется нейросетевая модель на основе многослойного перцептрона. На вход подается конкатенация векторов, описывающих сравниваемые модели, а на выходе формируется сигнал, значение которого интерпретируется в рамках шкалы классов качества. При наличии достаточного объема данных возможно достичь большей точности по сравнению с методом К-ближайших соседей, использующим взвешенные локальные метрики.

В работе [76] для сравнения прецедентов используются сиамские нейронные сети. На первом этапе входные векторы преобразуются в эмбединги — вектора, отражающие ключевые признаки сравниваемых ситуаций. Затем на выходном нейроклассификаторе вычисляется значение функции схожести между этими эмбедингами. Исследование охватило примеры из области принятия решений в индустрии аквакультуры, и эксперименты продемонстрировали высокую точность при сравнении и отборе прецедентов для систем CBR.

Одной из ключевых задач является формализация объекта с учетом его контекста и текущей ситуации. Это требует структурирования описания

возможных ситуаций таким образом, чтобы можно было проводить их сравнение и выбирать наиболее подходящие аналоги. Для достижения этой цели необходимо определить основные параметры описания, разработать структуру для представления типовых ситуаций и их взаимосвязей, а также организовать базу знаний, которая обеспечит быстрый поиск и извлечение аналогичных ситуаций.

Требования к структурированию и формализации знаний зависят от специфики предметной области. Например, важно учитывать, какая информация используется для описания атрибутов сложной ситуации, а также сколько времени и усилий потребуется для формализации исходных данных в условиях возникновения нежелательной ситуации.

Задача формализации объекта и ситуации на нем является ключевой для дальнейшего моделирования.

В этой области известны исследования, посвященные формализации, моделированию технологических объектов, следующих авторов: А.Ф. Антипин, В. К. Битюков, В. С. Кудряшов, М. В. Алексеев, Д. П. Лащенко, В. Л. Бурковский, Т. Н. Караневская, А. Г. Шумихин. В работах рассматриваются подходы к моделированию технологических объектов на основе показателей датчиков, построения уравнений теплообмена, [82, 83]. Известны работы моделирования объектов в целях ситуационного управления на основе семиотического подхода [84], моделирование на основе нечёткой логики с использованием многомерных интервально-логических регуляторов [85]. Однако данные подходы не предлагают учет условий контекста, что в случае сложного технологического объекта может иметь весомое значение.

Еще одна задача связана с проблемой отсутствия в базе подходящих решений для новой ситуации. Такая ситуация возникает, если в базе знаний отсутствуют достаточно схожие ситуации. Для таких условий предусматривается механизм адаптации уже имеющихся в базе решений [86], чтобы система смогла предоставить пользователям наиболее применимые решения.

Это особенно актуально для систем поддержки принятия решений в проблемных ситуациях на сложных технологических объектах. Здесь наряду с

высокой ответственностью и сложностью программ действий наблюдается значительная уникальность ситуаций. Несмотря на длительную историю развития CBR, поиск методов адаптации решений остается актуальным направлением исследований при разработке прикладных CBR-систем [87-89].

1.5 Задача классификации состояний элементов сложного технологического объекта

Отдельное внимание к данной задаче обусловлено тем, что элементы сложного технологического объекта могут иметь разнообразные характеристики и описываться различными наборами параметров. Для определения их состояний могут использоваться различные методы. В самом простом случае элемент может находиться в одном из двух состояний, таких как «Исправен» или «Неисправен», а также «Доступен» или «Недоступен». Однако источники данных для определения конкретного состояния могут различаться: это могут быть как табличные качественные и количественные данные, так и изображения с камер наблюдения.

Для анализа таких входных данных и их преобразования в математическое представление необходим классификатор, который выбирается в зависимости от типа поступающей информации. Классификация состояния элемента может основываться на данных, получаемых в реальном времени с датчиков мониторинга и телеметрии, что характерно для технологического оборудования, интерфейсов и линий связи.

Кроме того, классификация может проводиться на основе анализа косвенных признаков или другой качественной информации. Этот подход особенно актуален для систем операционного окружения, где данные в реальном времени не собираются напрямую, но могут быть получены по дополнительному запросу или из сторонних баз данных корпоративных информационных систем. Например, это может включать информацию о состоянии сервисных служб, наличии ресурсов и объектах инфраструктуры.

В рамках диссертации предполагается рассматривать два типа входных данных: табличные (качественные и количественные) и нетабличные (изображения). Однако предложенный подход допускает обработку и других видов данных, таких как видео, звук или текст.

Качественные и количественные табличные данные.

Когда имеется возможность собирать данные и формировать обучающие выборки для распознавания состояний, рекомендуется применять методы машинного или глубокого обучения [90, 91]:

Табличные данные: качественные и количественные.

При наличии возможности собирать данные и формировать обучающие выборки для распознавания состояний элементов сложных технологических объектов (СТО) рекомендуется использовать методы машинного или глубокого обучения. Эти подходы позволяют выявлять закономерности и строить модели, способные эффективно классифицировать состояния объектов.

Полносвязные искусственные нейронные сети.

Полносвязные нейронные сети требуют на вход нормализованные или классифицированные количественные табличные данные. Эти модели способны выявлять сложные зависимости и закономерности, однако их использование связано с определенными недостатками. В частности, для достижения высокой точности требуется значительное количество обучающих примеров, которые должны охватывать все возможные вариации данных.

Регрессионные модели.

Регрессионные модели представляют собой универсальные инструменты, которые описывают зависимости между параметрами. Они особенно хорошо работают с количественными табличными данными, где значения параметров представлены в числовом виде. Однако их применение имеет свои ограничения. Например, такие модели могут испытывать трудности при учете сложных нелинейных зависимостей между параметрами, что требует дополнительной интерпретации или уточнения выходных значений.

Метод К-ближайших соседей (KNN).

Метод К-ближайших соседей представляет собой классический подход к классификации объектов или ситуаций, описываемых набором нормализованных параметров. В этом методе расстояние между ситуациями вычисляется с использованием евклидовой метрики. Например, в задаче распознавания состояний объектов системы водоснабжения, описанной в работе [92], были проведены эксперименты, которые показали, что полносвязные нейронные сети обеспечивают более высокую эффективность по сравнению с классическим методом К-ближайших соседей.

Деревья решений.

Для работы с качественными табличными данными, представленными в символьной форме, деревья решений являются одним из наиболее подходящих инструментов. Они позволяют классифицировать данные, основываясь на последовательных разветвлениях, которые отражают ключевые признаки. Однако деревья решений имеют свои недостатки: они склонны к переобучению и ограничены в количестве входных параметров. Если число параметров превышает 5, качество классификации может значительно снизиться. Для устранения этих проблем и повышения точности часто применяются ансамблевые методы, такие как случайный лес, который объединяет множество деревьев решений для получения более стабильных и точных результатов.

Классификация на основе экспертных правил.

Если элементы сложного технологического объекта (СТО) характеризуются небольшим количеством параметров (до 3 количественных или качественных) и имеют ограниченное число возможных классов состояния (до 3), рационально применять методы, основанные на экспертных знаниях. Такие системы используют продукционные правила, которые воспроизводят процесс рассуждения эксперта. Этот процесс начинается с анализа исходных наблюдаемых данных или дополнительно запрашиваемой информации и завершается подтверждением гипотезы о состоянии элемента СТО.

Методы, основанные на экспертных знаниях, особенно актуальны в ситуациях, когда отсутствует достаточный объем обучающих данных, что часто наблюдается на начальных этапах эксплуатации объекта.

Классификация на основе экспертных правил.

Когда элементы СТО описываются малым числом параметров (до 3 количественных или качественных) и имеют ограниченное количество классов состояния (до 3), целесообразно использовать системы, основанные на продукционных правилах. В таких системах воспроизводится логика эксперта: от анализа исходных данных или дополнительно запрашиваемой информации до подтверждения гипотез о состоянии элемента СТО.

Архитектура графа вывода, основанная на продукционных правилах, представлена на рисунке 1.1. В этом графе каждая дуга, соединяющая две вершины, соответствует правилу формата «Если... То...». Исходная вершина отражает факт, проверяемый в условии «Если», тогда как конечная вершина представляет заключение, выведенное из данного правила. Каждый путь от входных вершин к конечным вершинам графа формирует независимую цепочку вывода, основанную на различных данных, что позволяет определить состояние элемента.

Реализация правил в виде продукций с коэффициентами возможности или в форме нечетких высказываний с использованием нечетких переменных позволяет получать итоговые оценки состояний (μ_k/S_k), где k обозначает индекс состояния. Для дальнейшего использования в формате one-hot кодирования производится пересчет для каждого элемента по правилу $\mu_k^* = 1$ при условии, что $\mu_k^* = \operatorname{argmax}(\mu_k)$, в то время как остальные значения μ_k устанавливаются в 0.

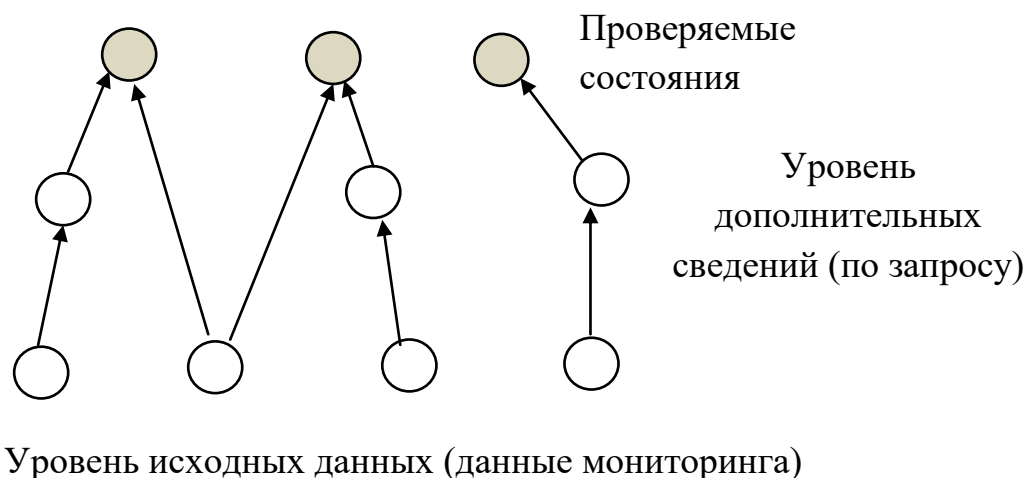


Рисунок 1.1 – Архитектура графа вывода при распознавании состояний

Аналогичным подходом дерево ситуаций [93]. В основе дерева ситуаций лежит графическая модель, где каждая ветвь символизирует возможный выбор или событие, а конечные узлы отображают результаты этих решений. Дерево ситуаций, представленное на рисунке 1.2, демонстрирует иерархическую структуру, основанную на отношениях «Вид-подвид». На нижнем уровне подвид S_{n+1} наследует атрибуты своего предшественника S_n и дополняет их новыми характеристиками, что позволяет уточнять и детализировать ситуации.

При движении по дереву сверху вниз, принимающее решение лицо уточняет значения атрибутов на каждом уровне. В нижней части дерева формируется описание комплексной ситуации, которое соответствует набору примеров из базы знаний и их решений — прецедентов. Это позволяет более точно анализировать и принимать обоснованные решения на основе имеющихся данных.

Нетабличные данные. Одной из распространенных форм данных, которые циркулируют в системах мониторинга и наблюдения, является изображение и видеоданные. Видеопоток для анализа на практике представляется дискретным набором кадров, то есть изображений, поэтому далее будем рассматривать именно работу с изображениями. Целесообразно применить сверточную нейронную сеть.

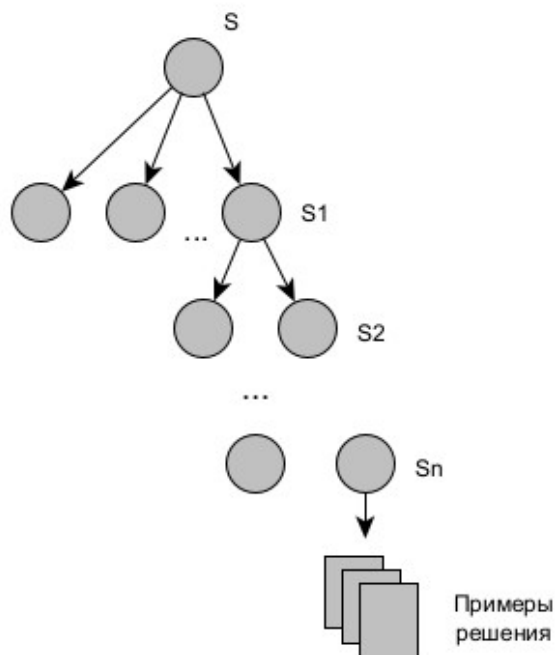


Рисунок 1.2 – Общий вид дерева ситуаций

Сверточные нейронные сети показали себя наиболее действенным инструментом для работы с изображениями [94]. В рассматриваемых задачах на практике состояние какого-либо из элементов СТО может контролироваться с помощью видео-, фото- фиксации. Таким образом, ставится задача бинарной или множественной классификации с помощью сверточной нейронной сети.

Типичная архитектура такой сети включает ряд слоев (рисунок 1.3) – сверточные слои с выделением ключевых признаков объекта и полносвязные слои для классификации. На вход сети подаются изображения (точнее, векторы, кодирующее представление изображений с помощью матриц пикселей), на выходе устанавливаются решающие элементы – нейроны, каждый из которых соответствует своему из классов. В случае бинарной классификации (например, распознавание состояний из пары «Исправен/Неисправен») решающий нейрон делается один, но значение на его выходе – 1 или 0 будет сигнализировать о том или ином состоянии рассматриваемого объекта.

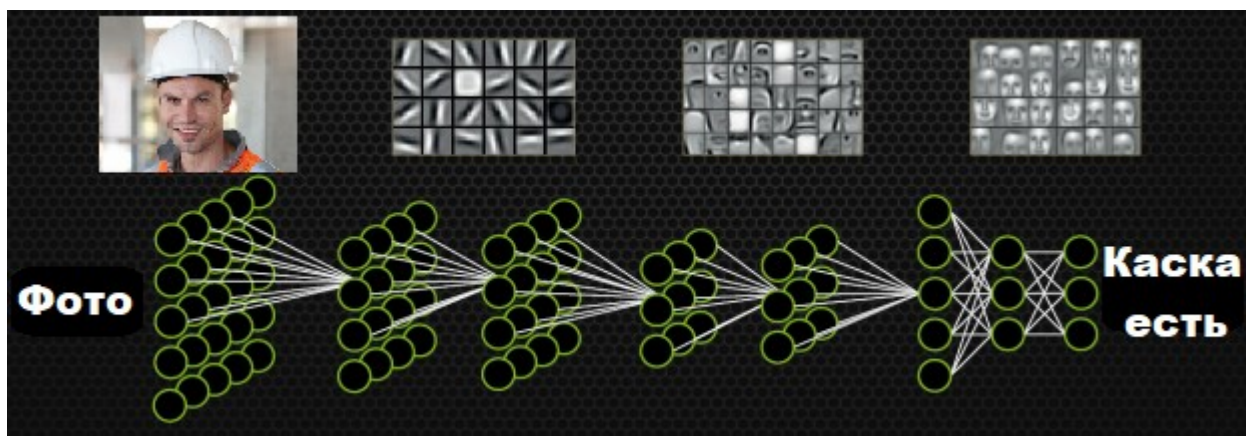


Рисунок 1.3 – Схематическое изображение нейронной сети для классификации изображений [95]

Таким образом, классификатор на основе сверточной нейронной сети становится одним из элементов множества классификаторов, который будет работать с изображениями.

Может использоваться для ряда типовых задач:

- Контроль людей и транспорта на стройплощадке;
- Контроль наличия средств индивидуальной защиты;
- Присутствие посторонних людей на территории;
- Контроль аварийного подъезда (свободен или нет);
- Контроль взаимного расположения тех. объектов и др.

Выводы по главе 1

В современных работах в области мониторинга технологических процессов и объектов большая часть работ посвящена проблеме сбора в реальном времени первичных данных, полнота и точность которых позволит делать вывод о состоянии объекта мониторинга.

В то же время вопросы оперативного принятия решений при возникновении или угрозе возникновения нештатных, опасных или аварийных ситуаций на уровне технологии остаются не в полной мере исследованными и реализованными. Это связано с высокой трудоемкостью задач моделирования

объектов (цифровых двойников), создания алгоритмов и баз знаний для прикладных ИСППР в каждой конкретной системе.

Анализ литературы позволил выделить основные научно-технические проблемы:

- сложность и высокая ответственность задач оперативного принятия решений в условиях возникновения нештатных, опасных ситуаций на сложных технологических объектах (на предприятиях городской обеспечивающей инфраструктуры, производственных и эксплуатирующих организациях);
- высокая трудоемкость и наукоемкость задач создания баз знаний и инструментария прикладных ИСППР, что тормозит их создание и внедрение в реальных условиях;
- научно-техническая проблема создания нейросимвольного интеллекта для возможности создания ИСППР, интегрирующих методы машинного обучения и инженерии знаний для работы в условиях малых объемов обучающих данных при возникновении редких или уникальных ситуаций.

Актуальным является создание программно-технического инструментария, который позволяет избежать выполнения сложных исследовательских работ в конкретной области и позволит выполнять "быструю сборку" прикладных ИСППР на основе имеющихся модулей, универсальных моделей, алгоритмов и предметно-независимых компонентов баз знаний.

Для реализации такого инструментария необходимо создание моделей и алгоритмов, которые позволили бы решить ключевые задачи, обозначенные в разделе 1.4 без привязки к предметной области.

ГЛАВА 2. МАТЕМАТИЧЕСКОЕ И ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ

2.1 Концептуальное моделирование сложного технологического объекта городской инфраструктуры

В качестве сложных объектов городской инфраструктуры авторы выделяют следующие системы: электроснабжение, водоснабжение (холодная вода), газоснабжение и теплоснабжение. У представленных сложных объектов основная функция – доставить ресурс от источника к конечному потребителю. Таким образом, системы осуществляют жизнеобеспечение города.

Такие сложные объекты представляют собой первоочередной интерес внедрения систем поддержки принятия решения.

Для формирования общей концептуальной модели технологического объекта рассмотрим эти системы подробнее.

Электроснабжение. Ресурс – электроэнергия. Источником электроэнергии для города выступает центр питания. Это может быть районная электростанция или понижающая подстанция глубокого ввода. Далее электроэнергия распределяется по городской сети через понижающие трансформаторные подстанции, установленные в разных частях города, и распределительные устройства, распределяющие электроэнергию по дворам и кварталам [96].

Концептуальная схема системы энергоснабжения города представлена на рисунке 2.1.

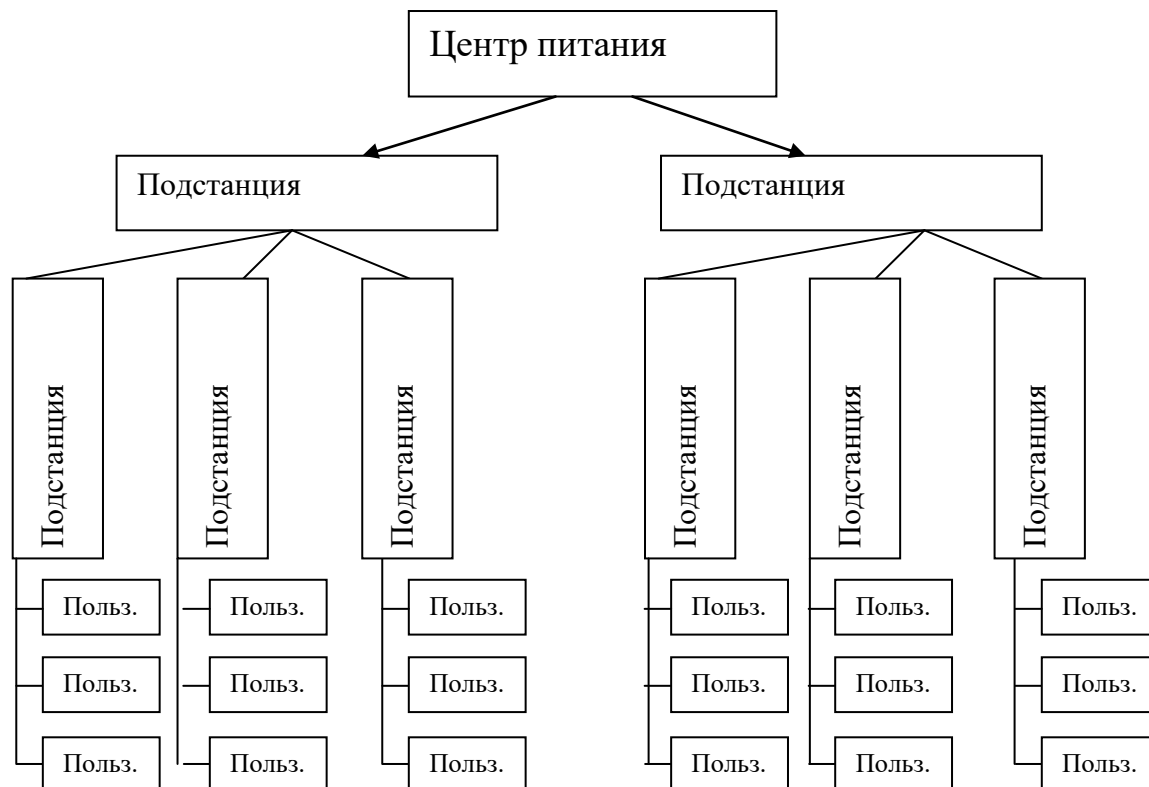


Рисунок 2.1 – Концептуальная схема системы энергоснабжения города

Водоснабжение. Ресурс – холодная вода. Источником в системе водоснабжения является водозабор из подземного или наземного водного объекта. Всю систему можно разделить условно на два уровня: уровень забора и подготовки воды и уровень городской сети водоснабжения. Первый уровень более сложный с технической точки зрения, включает в себя станцию водозабора, насосы, станцию водоподготовки, резервуары. Уровень городской сети представляет собой трубопроводы, по которым вода доходит к конечным потребителям, и дожимные насосные станции [97].

Концептуальная схема системы водоснабжения города представлена на рисунке 2.2.

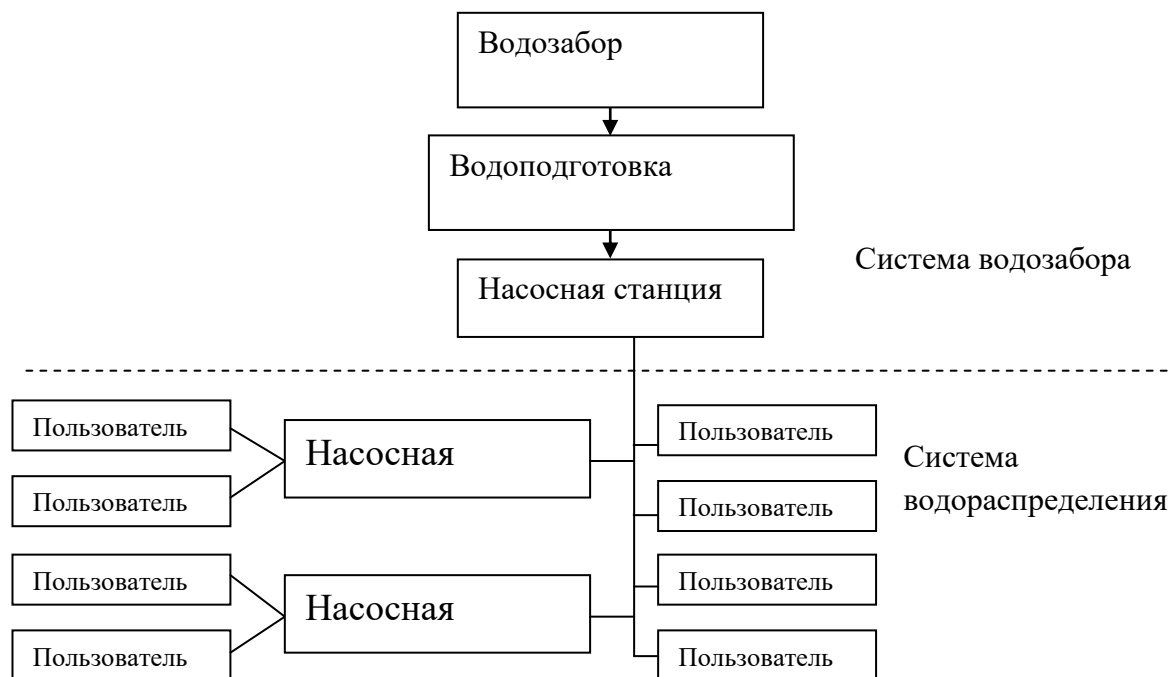


Рисунок 2.2 – Концептуальная схема сложного технологического объекта «водоснабжение»

Газоснабжение. Ресурс – природный газ. Источник – магистральный межрегиональный газопровод. Систему можно разделить условно на 3 уровня в зависимости от давления в трубопроводе. Газ, проходя через газорегуляторный пункт (ГРП), попадает на более низкий уровень с меньшим давлением. Таким образом, природный газ с постепенным снижением давления попадает из магистрального трубопровода к конечным потребителям [98].

Концептуальная схема системы газоснабжения города представлена на рисунке 2.3.

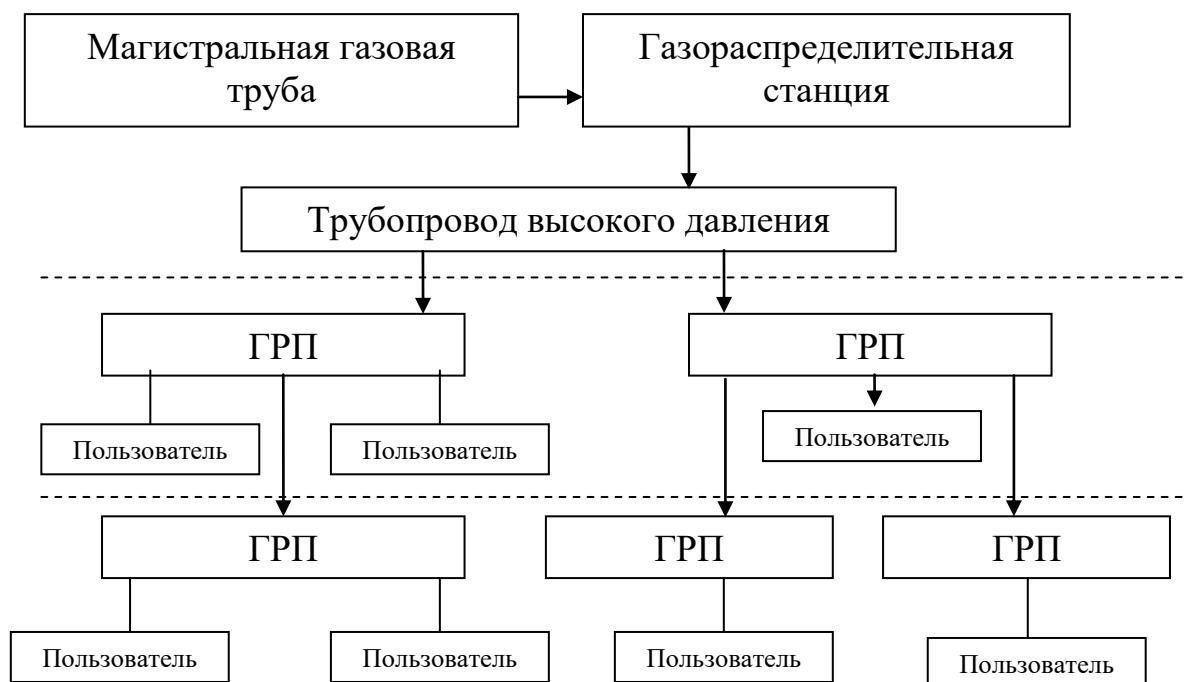


Рисунок 2.3 – Концептуальная схема сложного технологического объекта «газоснабжение»

Теплоснабжение. В рамках данной работы целесообразно рассмотреть именно централизованную систему теплоснабжения, которая функционирует в объемах квартала или целого города. Ресурс – горячая техническая вода (система отопления) и питьевая вода (горячее водоснабжение). Источником выступает тепловая станция. В системе также могут быть тепловые распределительные пункты и дожимные станции [99].

У всех систем городской инфраструктуры имеется общее сходство – от источника ресурс распределяется к потребителям, проходя через различные технологические узлы (насосная станция, ГРПШ, подстанция), которые в свою очередь являются сложными технологическими объектами и могут быть рассмотрены как отдельные целевые системы. Окружающая среда влияет на эксплуатацию, на процессы обслуживания объекта [100-102].

Каждый сложный технологический объект, независимо от принадлежности к системе инфраструктуры, состоит из различного оборудования, обслуживается персоналом и находится под различным воздействием окружающей среды.

Другими словами, сложный технологический объект функционирует в условиях некоторого *контекста*, который может влиять на возникновение ситуации на объекте и на состав принимаемого решения.

Рассмотренные объекты стационарно расположены, определены режимы эксплуатации, порядок эксплуатации. Объекты не возникают спонтанно. Должны быть описаны штатные и нештатные ситуации.

Следовательно, описанную концепцию можно распространить на другие объекты, которые удовлетворяют таким требованиям. В том числе, объекты производств.

2.2 Представление сложного объекта и ситуаций на сложном объекте

Структура сложного технологического объекта содержит в себе элементы различного типа. Представить сложный технологический объект (СТО), его элементы и связи между ними позволит онтологическая модель СТО. Модель отобразит компоненты объекта для учета и идентификации ситуации. При необходимости в модель добавляются связи и элементы.

На рисунке 2.4 представлена онтологическая модель сложного технологического объекта.

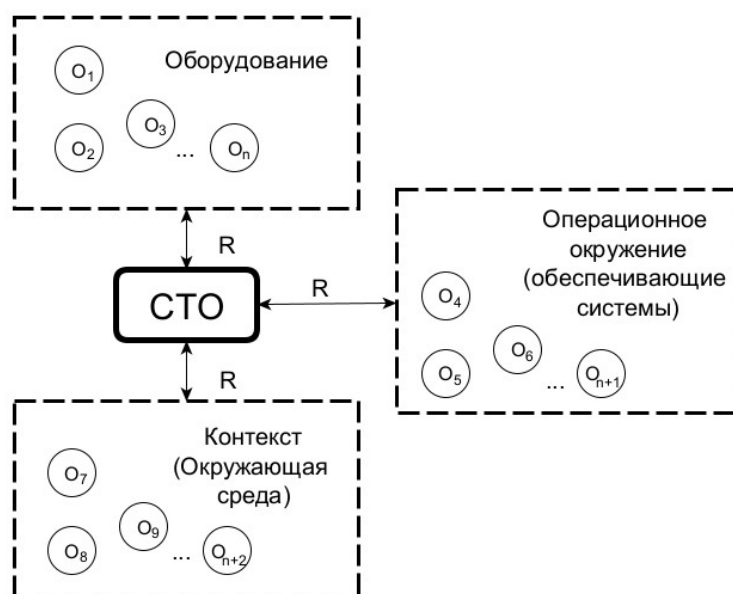


Рисунок 2.4 – Онтологическая модель сложного технологического объекта

Модель СТО может носить условно фрактальный характер, когда элемент при более детальном рассмотрении так же является сложным технологическим объектам с ранее обозначенными составляющими.

В онтологическом представлении сложный объект СТО описывается

$$\langle O, S, R \rangle, \quad (2.1)$$

где $O = \{O_i \mid i = 1, 2, \dots, N\}$ - множество элементов сложного объекта, которые принадлежат описанным выше подмножествам,

$S = \{S_j \mid j = 1, 2, \dots, M\}$ – множество возможных состояний.

R – множество отношений между элементами сложного объекта

$$R = \{R_k \mid \forall k \in K\},$$

где K – множество индексов отношений между элементами СТО.

Множество индексов отношений содержит типовые отношения Part-of, Has-a, Kind-of, etc. Предполагается, что могут быть добавлены специфичные отношения для определенного объекта;

Модель СТО включает не пересекающиеся подмножества, характеризующие технологические компоненты (Оборудование, Операционное окружение) и контекст (окружающая среда, обеспечивающие сторонние системы, персонал):

- «Оборудование» – элементы технологического оборудования, состояние которых определяет функционирование самого технологического объекта, его способность выполнять назначенные функции с заданным качеством и требованиями;
- «Операционное окружение» – элементы организационных и организационно-технических систем, влияющие на эксплуатацию, обслуживание, ремонт или смену технологического оборудования;
- «Контекст» – элементы окружающей среды, инфраструктуры, которые могут влиять на состояния и действия оборудования или операционного окружения, но на которые влиять нельзя;

«Отношения» – связи между элементами предыдущих подмножеств замыкают онтологическую модель.

Для дальнейших задач моделирования прецедентов и их преобразований модель (2.1) может быть упрощена, если и отношения, и элементы рассматривать относительно того в каком состоянии они находятся без учета семантики отношений. Для принятия решения целесообразно перевести связи, влияющие на принятие решения, в компоненты СТО.

Переходя от представленной онтологической модели к математической будем рассматривать сложный технологический объект как совокупность элементов разной природы, которые могут находиться в различных состояниях S:

$$\text{СТО} = \langle O, S \rangle, \quad (2.2)$$

где $O = \{O_i \mid i = 1, 2, \dots, N\}$ - множество элементов сложного объекта, которые принадлежат описанным выше подмножествам,

$S = \{S_j \mid j = 1, 2, \dots, M\}$ – множество возможных состояний.

Каждый из элементов O_i названных подмножеств может находиться в некотором состоянии S_i , соответствующих тому или иному элементу:

$$O_i \rightarrow S_i,$$

где $S_i = (p_i^1, p_i^2, \dots, p_i^{m_i}) = \{p_i^k\}_{k=1}^{m_i}$, при $m_i > 1$ – число классов (возможных состояний) для каждого элемента O_i , $p_i \in \{0,1\}$ – степень уверенности принадлежности к классу (вероятность нахождения в определенном состоянии).

Для частного случая неопределенности $p_i \in [0,1]$.

Определение 1. Ситуация на сложном технологическом объекте есть совокупность состояний, в которых находятся элементы множества O.

$$\text{Sit} \rightarrow \check{S} = (S_1, S_2, \dots, S_N) = \{S_i^k\}_{k=1}^N$$

Длина вектора ситуации $\check{S} = \sum_{i=1}^N m_i$.

Таким образом, ситуация в пространстве состояний Sit характеризуется кортежем векторов S_i , образующим вектор \check{S} .

Рассмотрим пример сложного технологического объекта — «умная» городская ферма, которая представляет собой автоматизированную систему для гидропонного выращивания растений [103]. Программно-аппаратное обеспечение фермы позволяет контролировать и управлять параметрами: температура, влажность, освещенность, уровень CO₂, pH питательного раствора.

Представить ситуацию можно через матрицу состояний, где единица в столбце соответствует состоянию элемента. В таблице 2.1 представлен пример комплексной ситуации на «умной» ферме.

Таблица 2.1 – Представление ситуации на «умной» ферме

Sit ₁	Элементы и параметры	исправно	сломано	ниже порога	минимум	оптимум	максимум	выше порога
Система поддержки микроклимата	Температура			0	0	1	0	0
	Датчик температуры	1	0					
	Влажность			0	0	1	0	0
	Датчик влажности	1	0					
	Уровень CO ₂			0	0	1	0	0
	Датчик CO ₂	1	0					
Система полива	Ph (кислотность раствора)			0	1	0	0	0
	pH-метр	1	0					
	ЕС по элементный			0	1	0	0	0
	датчик электропроводности	1	0					
	Температура раствора			0	0	1	0	0
	Датчик температуры	1	0					
	Интенсивность полива			1	0	0	0	0
	Таймер полива	0	1					
Освещение	Освещение	1	0					
	Мощность освещения			0	0	1	0	0
	Периодичность включения освещения			0	0	1	0	0
Состояние растения	увядание				0	0	1	0
	пожелтение				0	1	0	0
	ожоги				0	1	0	0

Для удобства представления и анализа ситуаций элементы системы технологического объекта (СТО) и оцениваемые параметры умной фермы сгруппированы в следующие категории: «Микроклимат», «Питание», «Освещение», «Растения». Эти категории согласуются с ранее обозначенными подмножествами.

Вектор \check{S} ситуации представляет собой кортеж векторов S_i , где каждому S_i соответствует строка таблицы 2.1. Таким образом, \check{S} формируется путем конкатенации строк таблицы, для примера по данным таблицы 2.1 имеем следующее внутренне представление ситуации (Рисунок 2.5):



Рисунок 2.5 – Вектор \check{S} , описывающий ситуацию на умной ферме, представленную в таблице 2.1

2.3 Модели представления прецедентов в ситуационной базе знаний

Положим, что целью принятия решения при возникновении текущей проблемной ситуации Sit_{act} является перевод ее в целевую ситуацию Sit_{end} . Далее для упрощения мы полагаем, что целевая ситуация одинакова для всех возможных входных ситуаций и соответствует тому случаю, когда все элементы технологического объекта и связи между ними находятся в исправных, работоспособных состояниях.

В общем случае этот перевод Sit_{act} в Sit_{end} является многошаговым дискретным процессом смены состояний элементов. Допустим, что каждой смене элементарного состояния соответствует некоторое действие в системе.

Тогда последовательность мультивекторов $\check{S}^0, \check{S}^1, \dots, \check{S}^k, \dots, \check{S}^{k_{end}}$, где

$$\begin{aligned} \text{Sit}_{act} &\rightarrow \check{S}^0, \\ \text{Sit}_{end} &\rightarrow \check{S}^{k_{end}} \end{aligned}$$

будет однозначно представлять программу действий, преобразующих входную проблемную ситуацию в целевую.

Обозначим эту последовательность:

$$R^\gamma = (\check{S}^0, \check{S}^3, \dots, \check{S}^k, \dots, \check{S}^{k_{end}}), \quad (2.3)$$

где γ - как и ранее есть все множество значений индекса k и $\check{S}^{k-1} \neq \check{S}^k$ для всех k .

Таким образом, R^γ также является вектором с дополнительной осью измерений.

Мультивектор R^γ есть формальное представление решения Sol некоторой ситуации Sit^γ .

Определение 2: Решение – инструкция по переводу объекта из критической ситуации в целевую *исправную*, которая описывается последовательностью векторов R^γ .

Полагаем, что в базе прецедентов представлены последовательности R для некоторого количества m ситуаций Sit .

Итоговую модель представлений для пары <Ситуация, Решение> запишем следующими выражениями:

$$\begin{aligned} Sol &\rightarrow R^\gamma, \\ Sit &\rightarrow \check{S}, \\ \text{Case} &= \langle Sit, Sol \rangle \end{aligned} \quad (2.4)$$

Здесь в частном случае вместо общей Sit может быть как ситуация в базе знаний, так и входная проблемная ситуация.

Главное преимущество этой модели представлений состоит в том, что она позволяет от большого количества разнородных параметров, которыми могут описываться элементы O_i сложного объекта, и многообразия их типов перейти к единообразному описанию ситуаций на сложном объекте и решений для управления этим объектом.

Иллюстрация модели представления для случая с двумя элементами сложного объекта приведена на рисунке 2.6.

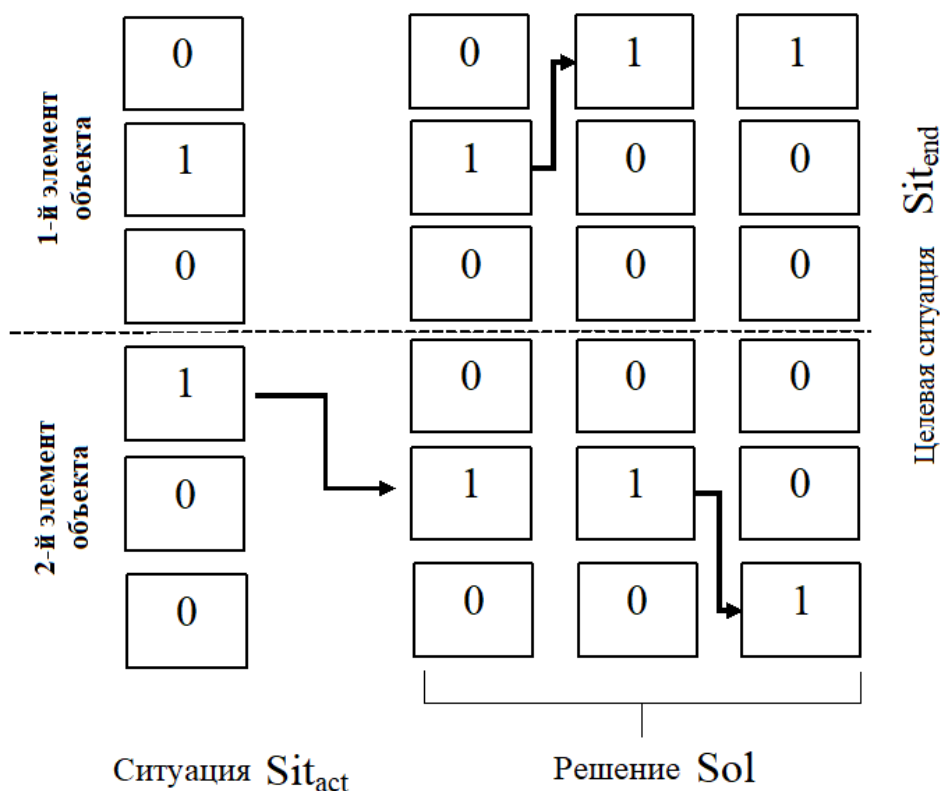


Рисунок 2.6 – Представление ситуаций Sit_{act}, Sit_{end} и решения Sol

Эта модель также стала основой для нового подхода к выбору ситуаций в базе знаний ИСППР, который описан далее в разделе 2.4.

Рассмотрим упрощенный пример – аварийная ситуация на понижающей электроподстанции. В данном случае наблюдается воздействие природных объектов: упавшее дерево повредило передаточный узел сервера связи подстанции, в результате чего была утеряна связь с устройствами телемеханики.

Приведение аварийной ситуации к исправной целевой можно представить в виде диаграммы смены состояний UML (Рисунок 2.7), которая является отображением траектории в пространстве состояний. При этом параметры входной ситуаций Sit_{act} представлены в верхней строке.

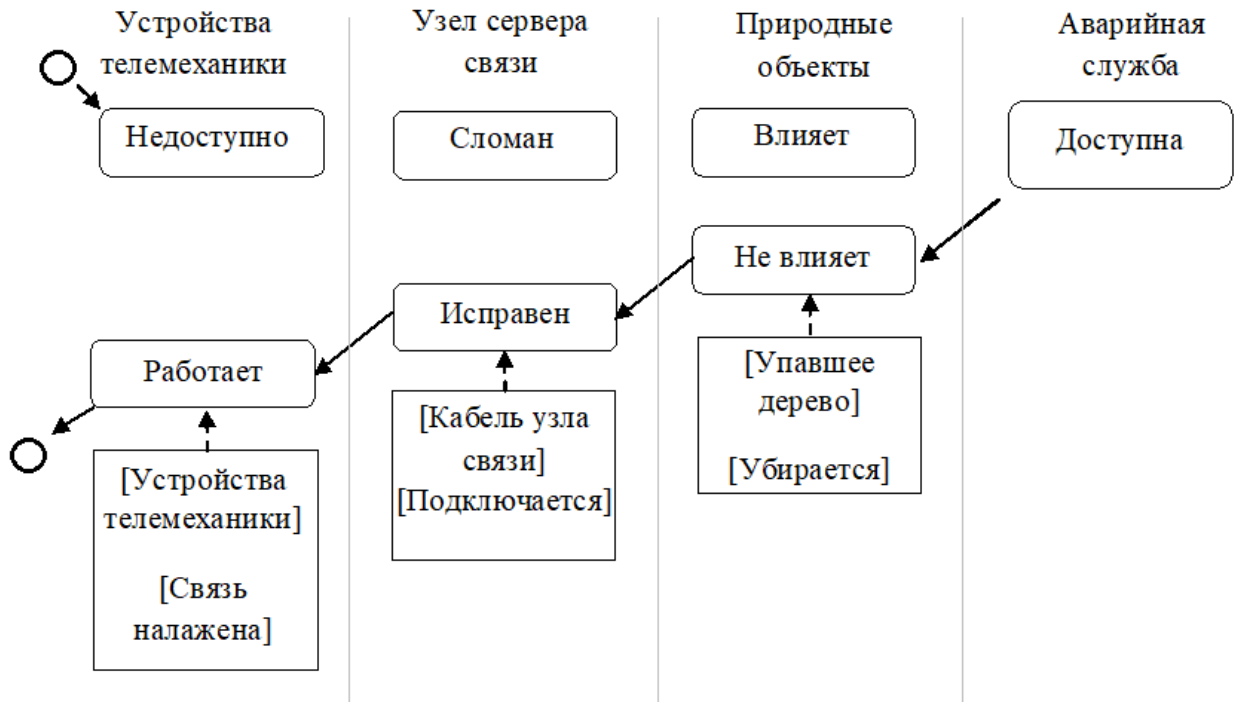


Рисунок 2.7 – Диаграмма UML смены состояний

Таким образом, при состоянии аварийной службы «доступна» производится последовательная смена состояний на желаемые, при которых достигается целевая ситуация – исправная работа подстанции.

2.4 Моделирование процесса выбора решения

Гибридный метод CBR, в котором используется как инженерия знаний, так и машинное обучение можно представить в виде схемы (Рисунок 2.8).

На рисунке 2.8. используются следующие обозначения:

O_i – i -тый элемент СТО;

N – количество элементов на объекте;

X_i – информация, поступающая от элемента O_i ;

K_i – функция определения класса состояния i -го элемента (классификатор из репозитория моделей машинного обучения и инженерии знаний);

$S_{i \text{ act}}$ – вероятностный вектор актуального состояния элемента O_i ;

\check{S}_{act} – вектор текущей ситуации;

\check{S}_z – вектор z -й ситуации в базе знаний;

$f(\check{S}_{act}, \check{S}_z)$, возвращающей количественную оценку схожести схожести ситуаций;

Sim – значение схожести.

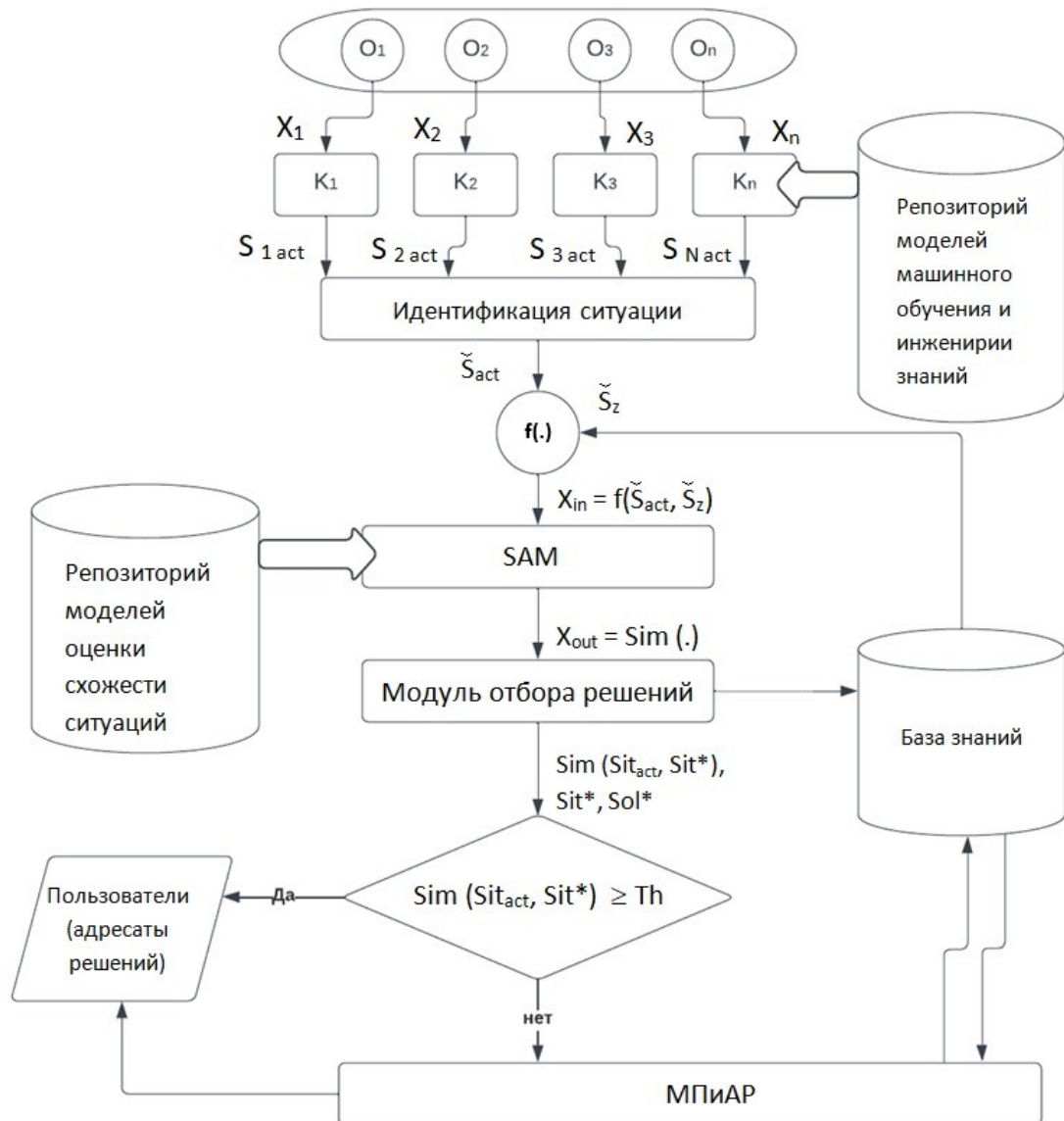


Рисунок 2.8 – Схема работы гибридного CBR

Реализация метода включает в себя четыре этапа.

1 этап. Сбор и анализ данных.

Цель этапа: формализация разнородных данных, получение вероятностного вектора состояния каждого элемента.

Согласно данному подходу в процессе эксплуатации сложного технологического объекта проводится отдельный сбор и анализ разнородных

данных, поступающих от его элементов. Конечной целью анализа является классификация состояния элемента путем определения вероятностей принадлежности к классу состояния.

Для каждого элемента O_i используется свой классификатор K_i из репозитория моделей классификаторов, Классификатор возвращает вероятность принадлежности классам состояний, что и определяет вероятностный вектор состояния $S_{i \text{ act}}$.

$$K: X_i \xrightarrow{K_i} S_{i \text{ act}},$$

где $S_{i \text{ act}} = (p_{i1}, p_{i2}, \dots, p_{im_i}) = \{p_i^k\}_{k=1}^{m_i}$, при $m_i > 1$ – число классов (возможных состояний) для каждого элемента O_i , $p_i \in [0,1]$ – вероятность принадлежности к классу (вероятность нахождения в определенном состоянии).

При этом в зависимости от имеющихся данных и степени неопределенности в качестве классификатора могут использоваться как модели машинного обучения, так и функции, основанные на экспертных знаниях:

- при наличии количественных данных и достаточном числе обучающих примеров – классификаторы из арсенала машинного или глубокого обучения (регрессионные модели, деревья решений, нейронные сети и др.);
- при недостатке обучающих данных и преобладании качественных данных – модели классификации на основе правил (экспертные правила, системы нечеткого вывода и др.).

2 этап. Идентификация ситуации.

Цель этапа: классификация комплексной ситуации для выявления нежелательной ситуации (требует принятия мер).

Вектора актуальных состояний формируют вектор актуальной комплексной ситуации \check{S}_{act} на объекте:

$$S_{it} \rightarrow \check{S}_{\text{act}} = (S_{1 \text{ act}}, S_{2 \text{ act}}, \dots, S_{N \text{ act}}) = \{S_{i \text{ act}}^k\}_{k=1}^N \text{ длиной } \sum_{i=1}^N m_i$$

Определение класса ситуации:

$$s: \check{S}_{\text{act}} \rightarrow C, \text{ где } C = \{0,1\},$$

s – функция модуля идентификации.

В зависимости от имеющихся данных, экспертных правил определения нежелательной ситуации в качестве функции s могут использоваться как модели машинного обучения, так и функции систем, основанных на экспертных правилах. С точки зрения практического использования допустимо ограничиться решением задачи бинарной классификации, это позволяет сразу отбросить ситуации, не требующие принятия мер.

3 этап. Отбор ситуаций из базы знаний.

Цель этапа: извлечь из базы знаний набор ситуаций в заданном количестве M , отранжированных по степени схожести Sim . Для этого вводится функция схожести f .

Используется модель оценки схожести (МОС), основанная на использовании функции f оценки схожести, возвращающей количественную оценку схожести ситуаций $Sim \in [0,1]$.

$$f(\check{S}_{act} \cdot \check{S}_z) = Sim,$$

где $\check{S}_z = \{S_{iz}\}_{i=1}^N$ – вектор, описывающий комплексную ситуацию z из БЗ при $\{z_i^k\}_{k=1}^{m_i}$, $z_i \in \{0,1\}$ – значение принадлежности к классу (нахождение в определенном состоянии).

Аналогично этапу 2 в зависимости от имеющихся данных и степени неопределенности могут использоваться для определения этой функции как техники машинного обучения, так и техники систем, основанных на знаниях:

- при количестве элементов более K^* , непостоянных весовых коэффициентов важности этих элементов, непостоянных весовых коэффициентов состояний элементов и достаточном числе обучающих примеров – *обучаемая функция выбора решения на основе схожести*;
- при недостатке обучающих данных и малом количестве элементов (менее K^*) – классические метрики оценки схожести: метрика Евклида, манхэттенская метрика, косинусная, метрика Тверского и др.

* Определяется экспертной оценкой

4 этап. Формирование решения по базе знаний.

Цель этапа: выдача решения адресатам.

С помощью модуля отбора решений из базы знаний выбирается наиболее сходная ситуация Sit_z (множество сходных) по критерию:

$$\text{Sim}(Sit_{act}, Sit_z) \rightarrow \max \quad (2.5)$$

При выполнении условия

$$\text{Sim}(Sit_{act}, Sit_z) \geq Th^* \quad (2.6)$$

ее решение из пары $\langle Sit^*, Sol^* \rangle$ выдается адресатам – пользователям системы.

Если это условие не выполняется, модуль поиска и адаптации решений (МПиАР) выполняет задачи преобразования решений из базы знаний для их использования в новой ситуации. Подробнее алгоритмы МПиАР рассмотрены в главе 3, раздел 3.2.

В итоге формируется цепочка – последовательность преобразований из заданной ситуации в конечную.

Выводы по главе 2

В результате концептуального моделирования сформулирована концепция моделирования сложного технологического объекта. Сложный технологический объект, независимо от принадлежности к системе инфраструктуры, состоит из различного оборудования, обслуживается персоналом и находится под различным воздействием окружающей среды. Такие объекты имеют следующие особенности: стационарно расположены, определены режимы эксплуатации, порядок эксплуатации. Объекты не возникают спонтанно, описаны штатные и нештатные ситуации.

Сложный технологический объект включает predetermined связи и состояния не только технологических компонентов, но и персонала, зданий и сооружений, подъездных путей и окружающей среды, которые являются элементами СТО.

* Определяется экспертной оценкой

Разработан метод математического моделирования ситуации на сложном технологическом объекте. Метод моделирования основан на представлении ситуации в виде совокупности состояний, в которых находятся элементы этого объекта. Разработана концепция моделирования решения. Решение представляет собой инструкцию по переводу объекта из критической ситуации в целевую *исправную*. В математическом виде инструкция является последовательностью векторов-ситуаций, смена которых описывается необходимыми действиями (технологической картой) и справочной информацией.

Представлен алгоритм гибридного-CBR и разработан численный метод процесса построения решения, включающая в себя 4 этапа: сбор и анализ данных, идентификация нежелательной ситуации (требует действий), отбор ситуаций из базы знаний, формирование решения из базы знаний.

Предлагаемый гибридный-CBR позволяет оперативно выводить решение в сложившейся ситуации на основе различных данных. При этом он компенсирует недостатки систем, основанных на знаниях (требуют трудоемкой работы по выявлению и формализации знаний), и систем на основе машинного обучения (требуют большие объемы обучающих примеров). Это достигается, в том числе, за счет применения разработанного метода моделирования ситуации на СТО и модели решения, которая позволяет переносить всю разнородную информацию в единое пространство состояний.

Результаты главы опубликованы в статьях [104-106].

ГЛАВА 3. РЕШЕНИЕ ЗАДАЧ ОТБОРА СИТУАЦИЙ И АДАПТАЦИИ РЕШЕНИЙ

3.1 Обучаемая функция выбора решения

Общая постановка задачи выбора решения.

В системах с выводом решения на прецедентах под прецедентом понимается пара Ситуация-Решение [107, 108]. При поступлении на вход системы текущей ситуации Sit , требующей своего решения, в базе прецедентов ищется наиболее схожая ситуация и потом выбирается то решение, которое составляет с ней пару. Таким образом, оценка схожести ситуаций является ключевой задачей.

Для количественной оценки схожести вводится так называемая Similarity-функция Sim [109], область значений которой лежит в интервале от 0 до 1. Если для двух ситуаций $Sit1$ и $Sit2$ найденное значение $Sim(Sit1, Sit2) = 1$, то это означает полную схожесть этих ситуаций. В частности, должно выполняться $Sim(Sit1, Sit1) = 1$, а $Sim(Sit1, Sit2) = 0$, напротив, означает полную несхожесть (в смысле принятого способа вычисления Sim).

Очевидно, что в данной постановке для заданной Sit в базе прецедентов нужно найти такую Sit^* , которая обеспечит наибольшее значение $Sim(Sit^*, Sit)$ по сравнению с остальными. Но на практике более целесообразно определить подмножество схожих ситуаций:

$$\Theta = \{Sit_z | z=1, 2, 3, \dots, Z\},$$

где Z – число ситуаций-примеров в БЗ такой Sit^* , что $Sim(Sit_{act}, Sit_z) > Th$,
 $Sim(Sit_{act}, Sit_z) \rightarrow \max$

Подмножество схожих ситуаций оставляет возможность выбора из нескольких прецедентов, окончательный выбор среди которых может осуществляться по дополнительным критериям.

Для оценки подобия ситуаций между собой в CBR более важным становится не их принадлежность к одному или другому классу, а некоторая

специальная функция схожести – similarity-функция [56], значение которой и будет показывать схожесть ситуаций между собой.

Вычисление similarity-function может быть выполнено на основе метрик – путем вычисления расстояния между ситуациями в пространстве атрибутов, которыми эти ситуации описываются.

Существует большое число метрик, которые применяются в CBR, метрика Евклида, манхеттенская метрика, косинусная, метрика Тверского и др. Чем более разнообразны параметры X и чем больше их количество, тем более сложной становится метрика.

Более точное вычисление Sim с помощью метрик обладает гораздо большей гибкостью, позволяет отвечать не только на вопросы вида «Схожа ситуация А с ситуацией В?», но и вопросы вида «Насколько ситуация А похожа на ситуацию В?» или «Какая из ситуаций В или С более похожа на А?». Таким образом, есть возможность отбора множества конкурирующих допустимых решений и дальнейшего выбора по дополнительным критериям (например, по критерию затрат на реализацию решения).

В случае сложных технологических объектов возникает ряд трудностей: требуется учет не только параметров самого оборудования, но и *контекста*.

Все это приводит к большому количеству параметров и потенциальному усложнению метрик. Таким образом, попытка сравнить ситуации в многомерном атрибутивном пространстве сложного объекта сопряжена со следующими проблемами:

- Необходимость создания локальных метрик схожести и их агрегирование в глобальные метрики;
- Необходимость экспертной оценки при ранжировании важности атрибутов для конкретной ситуации;
- Необходимость выявления *коллизий*, т.е. случаев, когда различие ситуаций по одним атрибутам (локальным метрикам) может компенсироваться их сходством по другим атрибутам.

Описанные проблемы связаны с трудоемкими задачами, которые требуют экспертного вмешательства. При этом вместе с усложнением объекта, возрастает трудоемкость таких задач и вероятность возникновения ошибок и коллизий.

Предполагается, что в случае сложного технологического объекта эффективным будет применение моделей машинного обучения. Это потенциально более гибкий подход, который позволит учесть трудноформализуемые знания экспертов с меньшими трудозатратами.

Методология решения.

Как упомянуто выше, одним из подходов к разработке Sim является применение методов машинного обучения [76, 81], когда модель функции Sim формируется на базе предъявляемых схожих и несхожих примеров, т.е. используется обучаемая модель Sim. Для проверки ее работоспособности требуется оценка качества модели.

Практика построения моделей машинного обучения и, в частности, нейросетей показывает, что требуется экспериментально подбирать параметры и архитектуры этих моделей с тем, чтобы найти то сочетание, которое обеспечит наилучшие метрики качества [110]. В случае значительного числа параметров, вариантов архитектур этот процесс становится трудоемким. Еще важнее, что эта сложность может привести к тому, что будет пропущена та модель, которая обеспечивает наилучшие результаты.

Таким образом, исходная задача сводится к автоматическому поиску лучших параметров нейросетевой модели для функции Sim на основе оптимизации заданных критериев качества.

Для ее решения вначале разрабатывается математическая модель, параметрами которой можно управлять, формируется набор критериев качества, потом описывается алгоритм поиска оптимальной модели.

Математическая модель similarity-функции.

Similarity-функция вводится для вычисления схожести в целях определения наиболее схожей ситуации и дальнейшего выбора прецедента и его решения.

Пусть две сравниваемые ситуации образуют вектор X_{in} , который будет подаваться в качестве входного параметра similarity-функции. Способы формирования X_{in} приводятся в работе [105]. Обобщенная математическая модель, отражающая последовательность преобразований входного набора данных в многослойной нейронной сети, может быть представлена следующим образом:

$$\text{Sim}(\text{Sit}_{act}, \text{Sit}_z) = f_n(W_n, (f_{n-1}(W_{n-1}, (\dots, f_1(W_1, X_{input}))))), \quad (3.1)$$

при $W = A(L, (y, y_{pred}))$
 $W \in R^{m_k \times d_k}$,

где W – обучаемые параметры (веса),

$\{f_p \mid p=1,2,3,\dots\}$ – множество функций активации,

$L \in \{L_p \mid p=1,2,3,\dots\}$ – множество функций потерь,

$n \in \{n_p \mid p=1,2,3,\dots\}$ – число слоев,

y, y_{pred} – метка истинная и предсказанная,

A – алгоритм обучения,

m_k – число нейронов на k -м слое,

d_k – размерность X на входе ($k = 1, \dots, n$),

$X_{input} = T(X_1, X_2)$, где X_1, X_2 – вектор входных ситуаций,

T – способ формирования X_{in} .

Обобщенная ансамблевая модель.

Добавим в (3.1) возможность ансамблирования моделей:

$$\text{Sim}(\text{Sit}_{act}, \text{Sit}_z) = F(\text{Agr}(f_{nj}(W_{nj}, (f_{nj-1}(W_{nj-1}, (\dots, f_{1j}(W_{1j}, X_{inputj}))))))), \quad (3.2)$$

где $j = [1; N]$, N – число моделей в ансамбле,

F – функция активации выходного слоя, $F \in \{F_p \mid p=1,2,3,\dots\}$,

Agr – функция агрегации, $\text{Agr} \in \{\text{Agr}_p \mid p=1,2,3,\dots\}$.

При $N=1$ ансамблевая модель становится обычной многослойной сетью.

Постановка задачи оптимизации архитектуры нейросетевой модели для вычисления similarity-функции.

Задача оптимизации состоит в том, чтобы подобрать управляемые параметры (гиперпараметры) модели (3.2) так, чтобы оптимизировать критерий качества нейросетевой модели.

Критерием оптимизации являются метрики качества. Если использовать одну из них, например, известную метрику для прогнозных моделей MAPE (Mean Absolute Percentage Error), то получаем однокритериальную задачу оптимизации:

$$(1 - K) \rightarrow \max,$$

где $K = K(\text{Sim}^\phi, \text{Sim}')$, Sim^ϕ – фактические значения для данных валидационного дата-сета $(X_{in}, \text{Sim}^\phi)$, Sim' – оценка similarity-функции на выходе нейросетевой модели.

Если использовать несколько метрик (например, [105, 110-112]), то получаем многокритериальную задачу:

$$(K_1, K_2, K_3, \dots) \rightarrow \max,$$

где потребуется вводить модель компромисса между критериями.

Предполагаем, что все метрики качества имеют одинаковую важность. Тогда многокритериальная задача сводится к однокритериальной путем аддитивной свертки равнозначных частных критериев K_p (предварительно они приводятся к одной шкале измерения – интервалу $[0,1]$ и одному направлению) с добавлением желаемых критериальных ограничений на минимально допустимые значения K_{\min_p} соответствующих метрик качества:

$$\sum_p K_p \rightarrow \max \quad (3.3)$$

$$K_p \geq K_{\min_p}$$

Решение этой задачи обеспечивается перебором управляемых параметров нейросетевой модели, которыми в (3.2) являются следующие:

- число моделей в ансамбле n , число слоев m_j в модели;
- размерность векторов T_{kj} на каждом k -м слое и в каждой j -й модели ансамбля;
- функция агрегирования $\text{Agr}_{j=[1:n]}$;

- функции активации, принимающие значения из множества доступных функций активации предполагаемой к использованию библиотеки Keras {Relu, Swish, Leaky Relu, ...}. Это множество может быть сокращено вплоть до одного элемента, если перебор данного параметра не требуется.

Обозначим вектор управляемых параметров как $Y = (y_1, y_2, \dots, y_R)$, где размерность вектора R есть общее число управляемых параметров.

Тогда решением (3.3) является такой Y^* , что $Y^* = \operatorname{argmax} (\sum_p K_p (Y))$ при соблюдении ограничений в (3.3).

Для решения этой задачи могут быть использованы методы поисковой оптимизации с эвристическим перебором значений управляемых параметров. Для сокращения множества перебора каждый из управляемых параметров y представляется перечислимым типом значений, т.е. ограниченным по размеру массивом (списком) возможных значений $y = [e_1, e_2, \dots, e_N]$. Ниже представлен алгоритм, который реализует один из известных методов поисковой оптимизации – метод покоординатного спуска (подъема).

Алгоритм оптимизации архитектуры нейросети.

- 1) Выбрать начальное значение Y //, например, вектор из первых элементов в списках возможных значений //
- 2) Обучить модель Sim на обучающем дата-сете
- 3) На вход модели подать наборы X_{in} из валидационного дата-сета
- 4) Вычислить значения метрик качества по прогнозам модели и, соответственно, критериев качества $\sum_p K_p$
- 5) Установить $r = 1$
- 6) Изменяя значение y_r делать шаги 2), 3), 4) пока не будет найдено наибольшее значение $\sum_p K$ и Y^*r //перебор останавливается в точке, после которой не происходит улучшения критерия качества с заданной точностью//
- 7) Установить $(r+1)$
- 8) Повторяем шаг 6)

9) Повторяем шаги 7), 8) до тех пор, пока $r \leq N$ (число управляемых параметров) или $\sum_r K$ перестает улучшаться.

10) За выходное значение Y^* принимаем точку Y^*_r , в которой останавливается алгоритм перебора.

Алгоритм был использован в работе [105] для исследования возможности определения схожести ситуаций в ИСППР на тепловом пункте здания.

В практике нейросетевого инжиниринга перебор большего числа вариантов часто позволяет найти ту модель, которая обеспечит значительное улучшение метрики качества, по сравнению с исходной моделью. Например, в работе [113] был исследован набор моделей, который потом был расширен и дополнен их ансамблями [105]. В число управляемых параметров включался дополнительно способ формирования входного вектора X_{in} по представлениям двух сравниваемых ситуаций. Расширение набора моделей за счет комбинирования управляемых параметров в исследовании [105] позволило получить новые варианты нейросетевых моделей для сравнения ситуаций, которые обеспечили более высокие показатели метрик качества на валидационных данных. Ограничением данного подхода будет возможно значительное время поиска оптимального варианта. Один цикл обучения–проверки нейросети в рассматриваемых задачах [105] длился в пределах 40 сек (оборудование: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 8 GB ОЗУ). Таким образом, в течение суток работы оптимизационного алгоритма может быть сгенерировано, обучено и проверено более двух тысяч вариантов. Это в любом случае будет значительно больше возможностей человека, который проводит подобную работу «вручную». Для сокращения объемов вычислений состав управляемых параметров Y и списки их допустимых значений предварительно устанавливается исходя из практики нейросетевого проектирования с учетом опыта и здравого смысла. Это позволяет сократить время решения оптимизационной задачи, однако, может привести к пропуску оптимального, возможно, неожиданного варианта.

Выводы по разделу 3.1

Многообразие элементов сложного и неоднородного объекта, а также их состояний, создает значительные трудности в эффективном отборе ситуаций. Это может привести как к высокой трудоемкости, так и к полной невозможности использования классических метрик для оценки расстояний в атрибутивных пространствах.

Для улучшения процесса оценки схожести ситуаций и их отбора активно применяются нейросети. Их использование позволяет избежать сложной задачи представления ситуаций в атрибутивных пространствах и минимизировать необходимость экспертной оценки при ранжировании важности атрибутов. Это также помогает устранить проблемы, связанные с возникновением коллизий, что в свою очередь ускоряет процесс принятия решений. Особенно это важно в критических ситуациях, где исключение человеческого фактора и снижение трудоемкости процессов принятия решений и разработки систем поддержки принятия решений (СППР) становятся ключевыми.

Исследования показали, что применение нейронных сетей в подобных сложных условиях является перспективным [62, 75, 77, 80]. Эти сети способны определять схожесть ситуаций с помощью машинного обучения на примерах пар схожих и несхожих ситуаций.

В рамках этого подхода был предложен метод автоматического поиска оптимальных параметров нейросетевой модели для similarity-функции, основанный на оптимизации заданных критериев качества.

Подход позволяет организовать упорядоченный перебор возможных архитектур нейросетевой модели для вычисления симилирити-функции и сделать это в автоматическом режиме. За счет этого достигается возможность сравнить большое число нейросетевых моделей и найти те, которые обеспечат лучшие метрики качества в условиях набора данных.

Подход позволяет автоматизировано подбирать параметры и архитектуры нейросетевых моделей с целью поиска сочетания, которое обеспечит наилучшие

метрики качества. В случае значительного числа параметров, вариантов архитектур подход позволяет существенно снизить трудоемкость разработки оптимальной нейросетевой модели. Более того, потенциально позволяет не пропустить модель, которая обеспечивает наилучшие результаты.

Результаты раздела опубликованы в работах [105, 112-115]

3.2 Алгоритм адаптации и синтеза решения

Постановка задачи: Поиск решений.

Эта задача формулируется исходя из гипотезы о том, что в длинных последовательностях преобразований (программах поэтапных действий для разрешения проблемной ситуации), известных в БЗ для одних ситуаций, могут присутствовать более короткие последовательности преобразований (программы действий), которые применимы для новой ситуации. В этом случае новая текущая ситуация Sit_{act} может рассматриваться как промежуточный этап преобразования уже известной ранее Sit в целевую Sit_{end} , описанного R^γ .

Задача поиска состоит в том, чтобы найти в множестве решений из БЗ такую связную часть последовательности вида (2.1), которая может быть принята в качестве решения текущей ситуации (3.4):

$$\check{S}_{act} \rightarrow \check{S}_z^k, \quad (3.4)$$

при $k > 0$ и $Sim(\check{S}_{act}, \check{S}_z^k) > Th$, $k = \operatorname{argmax}_{z \in \zeta, k \in \gamma} Sim(\check{S}_{act}, \check{S}_z^k)$.

Алгоритм решения данной задачи включает в себя перебор по z всех имеющихся в БЗ пар $\langle Sit_z, Sol_z \rangle$, выбор решения из каждой пары и поиск в последовательности мультивекторов R^γ этого решения такого компонента \check{S}^k , который наиболее сходен с \check{S}_{act} . На следующем этапе из множества найденных $\{\check{S}_z^k\}$ по критерию $Sim(\check{S}_{act}, \check{S}_z^k) \rightarrow \max$ выбирается лучший \check{S}_z^k , на основании которого далее восстанавливается вся последовательность преобразований, определяющих решение для данной ситуации R_{act}^γ . В таком случае $\check{S}_{act}^0 = \check{S}_z^k$.

На рисунке 3.1 контуром отмечена подпоследовательность $R^{\gamma 1}$. Приведение аварийной ситуации к исправной целевой ограничивается двумя последними шагами известной программы действий.

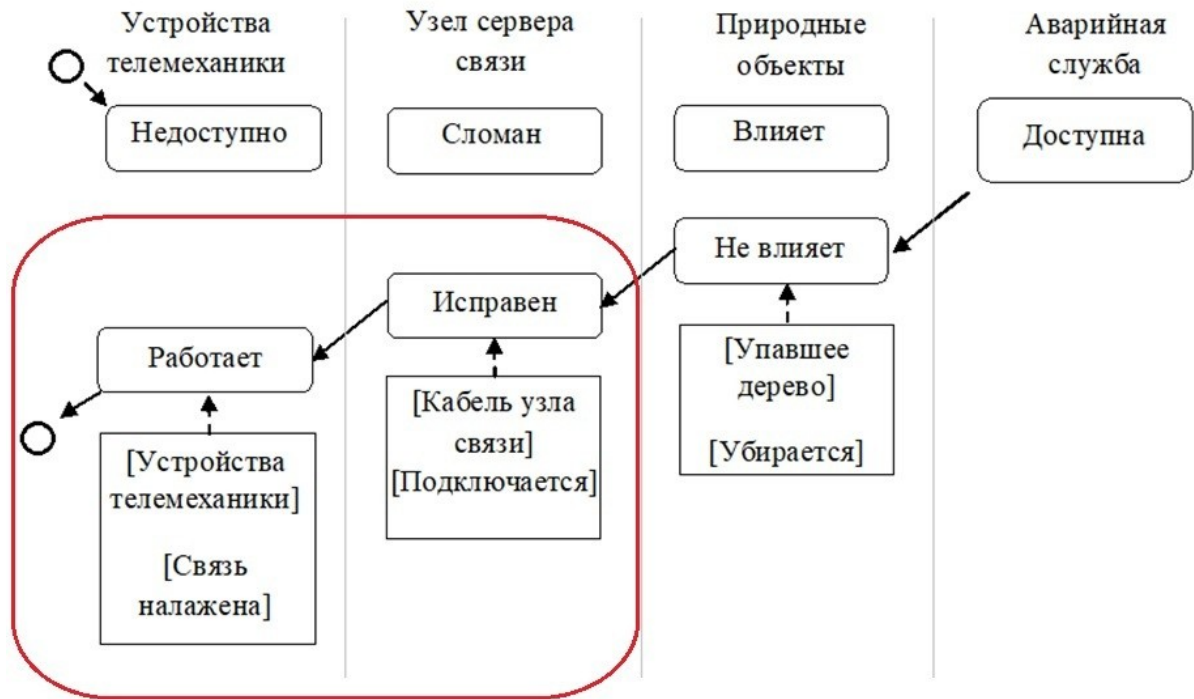


Рисунок 3.1 – Диаграмма UML смены состояний с отмеченной подпоследовательностью $R^{\gamma 1}$

Поэлементный синтез (сборки) решения.

Эта задача возникает в случае отсутствия в БЗ готовых решений или готовых подпоследовательностей, удовлетворяющих условию применимости (2.6). Она базируется на гипотезе о том, что при наличии некоторого решения (последовательности мультивекторов) им можно воспользоваться, если ситуацию предварительными преобразованиями привести к тому виду, который будет отвечать критерию допустимости (2.6), и эти преобразования могут быть найдены из других программ действий, имеющихся в БЗ.

Иначе говоря, в задаче 1 мы предполагаем, что в БЗ уже есть более длинные последовательности, связную подпоследовательность из которых можно использовать для разрешения текущей ситуации. В Задаче 2, напротив,

предполагается, что в БП последовательности преобразований более короткие, чем необходимые для разрешения данной ситуации.

Разложим \check{S}_z по элементам (3.5):

$$\check{S}_z = (S_{z1}, S_{z2}, \dots, S_{zN}) \tag{3.5}$$

где N – число элементов сложного объекта.

Положим, что в множестве решений в БП есть примеры элементарных преобразований (преобразования состояний отдельных элементов i), выполняемых за один дискретный шаг $Act^i: S_{zi}^s \rightarrow S_{zi}^{s+1}$.

На рисунке 3.2 контуром отмечено элементарное преобразование для первого элемента объекта.

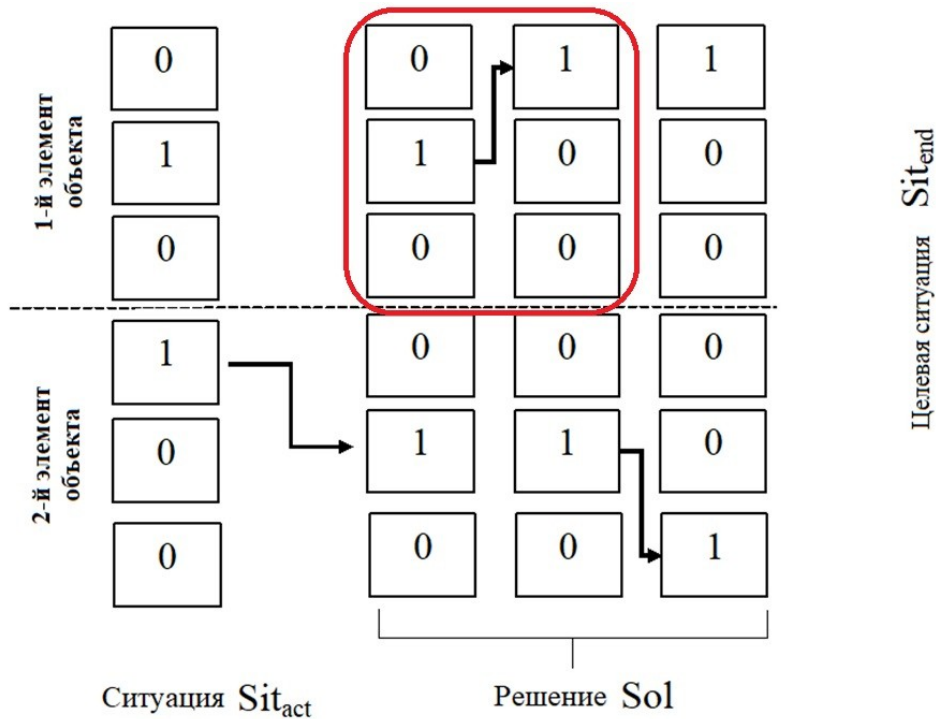


Рисунок 3.2 – Представление ситуаций Sit_{act} , Sit_{end} и решения Sol с отмеченным элементарным преобразованием 1-го элемента

Элементарные преобразования формируют последовательность элементарных преобразований для i элемента:

$$A^i: (S_{zi}^s, \dots, S_{zi}^{s_{end}}) \tag{3.6}$$

таких, что $\text{Sim}(S_{\text{act } i}, S_{z_i}^s) \geq \text{Th}$, т.е. эти вектора состояний обладают необходимой степенью схожести друг с другом, где s – некоторый номер в той или иной (z, i) -й последовательности состояний, соответствующей i -му элементу в z -й ситуации.

Тогда можно собрать последовательность R^α преобразований A^i , в которой поэлементно за N шагов производится перевод исходной ситуации i , соответственно, вектора \check{S}_{act} в желаемый \check{S}_z^k .

$$R^\alpha = (A^{i0}, A^{i1}, \dots, A^{i \text{ end}}) \quad (3.7)$$

Алгоритм для данной задачи является продолжением алгоритма выполнения задачи 1 и включается в работу после того, как принято решение о невыполнении условия применимости найденного на предыдущем этапе решения (т.е. найденного мультивектора \check{S}_z^k). На следующем этапе в БП путем перебора всех z для каждого i -го элемента находится пример элементарного преобразования такого, что его вход отвечает требованию схожести с соответствующим i -м мультивектором ситуации Sit_{act} . Далее эти преобразования соединяются в единую последовательность. На следующем шаге алгоритма полученная последовательность мультивекторов упорядочивается и трансформируется с учетом правил зависимости одних элементов от других, заданных в базе знаний системы.

На практике порядок смены состояний элементов может производиться с учетом некоторых условий зависимости одних элементов от других. Для отражения этого вводится система ограничений (аксиом), в соответствии с которой производится упорядочение элементарных преобразований.

Аксиомы – это ограничивающие комбинации элементарных преобразований Act^i и векторов состояний элементов S_k .

$$\langle \text{Act}^i, S_k \rangle \subset S,$$

где $k \in K$ – множество элементов, которые влияют на элемент i .

Ограничивающие комбинации задаются в виде матрицы смежности.

После учета аксиом формируется искомая последовательность состояний, где конечным элементом будет целевая ситуация $Sit_{end} \rightarrow \check{S}^{k_{end}}$.

Описанный алгоритм представлен на рисунке 3.3.

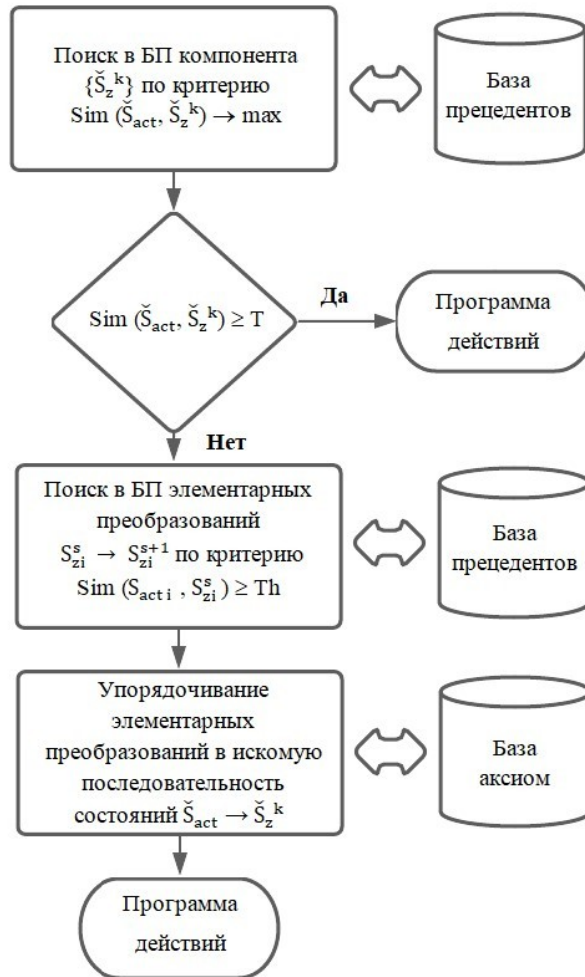


Рисунок 3.3 – Алгоритм поэлементного синтеза (сборки) решения

Пример аксиомы: недопустимая комбинация векторов, которая содержит следующую последовательность действий (изменение состояний элементов): Отключить выключатель → Вывести из работы устройство АПВ (автоматическое повторное включение). Такой порядок действий недопустим и опасен для жизни и оборудования при работах на электросетях. Следовательно, использование аксиомы позволит предотвратить недопустимую комбинацию и верно выстроить порядок действий при вновь собираемом решении.

Выводы по разделу 3.2

Предложенные алгоритмы и постановки задач значительно повышают надежность работы CBR-систем, позволяя находить решения в ситуациях, которые не охвачены базой прецедентов. Результаты этих исследований создают основу для дальнейшего развития методов адаптации решений и обновления базы прецедентов в процессе эксплуатации CBR-систем, учитывая выбранные модели представления. В частности, задача синтеза может быть усовершенствована для случаев, когда вместо простых преобразований в базе осуществляется поиск более сложных компонентов мультивекторов, состоящих из большего числа элементов. При этом могут вводиться дополнительные ограничения на сходство между мультивекторами элементов различных классов. Это может означать выделение элементов сложного объекта, состояния которых поддаются управлению, и тех элементов окружающего контекста, состояния которых остаются неизменными и должны демонстрировать обязательное сходство при сравнении ситуаций и выборе решений. Дальнейшие исследования в этой области будут сосредоточены на разработке новых формулировок задач и их применении для создания предметно-ориентированных систем поддержки принятия решений.

Описанный в разделе подход обуславливает архитектуру базы знаний. Состав решения формируется в виде отдельной таблицы, где каждая строка содержит шаг преобразования (вектор ситуации начальной, ситуации конечной), номер шага преобразования, связь с исходной ситуацией z и ссылка на технологическую карту, соответствующую этому шагу. Отдельная таблица содержит аксиомы.

Таким образом, возможна реализация поиска подходящей подпоследовательности, а с учетом информации о количестве элементов и их состояний появляется возможность поиска и извлечения элементарных преобразований и их увязка с технологической картой.

Результаты раздела опубликованы в работе [116].

ГЛАВА 4. КОМПЛЕКС ПРОГРАММ

Для реализации и тестирования представленных выше моделей и алгоритмов был разработан *комплекс программ*. На основе полученных результатов реализован программный комплекс «Платформа для быстрой сборки интеллектуальных систем поддержки принятия решений при эксплуатации сложных технологических объектов» далее – Платформа.

Под сборкой понимается настройка Платформы под конкретный сложный технологический объект путем подключения и дообучения программных модулей и нейросетевых моделей, которые обеспечивают процесс вывода решений пользователям согласно шагам схемы работы гибридного CBR (описаны в разделе 2.4).

Оригинальность программного решения определяется тем, что логическая и физическая архитектура Платформы реализует концепцию гибридного-CBR и идею быстрого создания системы поддержки принятия решений.

В приложении 3 представлено руководство пользователя программного комплекса.

4.1 Логическая архитектура Платформы

Логическая архитектура Платформы разработана на основе схемы гибридного ситуационного вывода (Рисунок 4.1) и предусматривает набор модулей и их взаимосвязь, которые обеспечивают следующие функции:

- обеспечение создания предметно-ориентированной интеллектуальной системы вывода решений на прецедентах в ситуациях, возникающих на сложных технологических объектах;
- сборка функциональных модулей в готовую систему вывода решений;
- заполнение и ведение базы правил логического вывода;
- ведение базы прецедентов в виде пар «Ситуация-Решение»;
- классификация состояний элементов сложного объекта по входным данным;

- вывод решения по входным данным о ситуации.

На рисунке 4.1 показана логическая архитектура Платформы. На вход от элементов поступают данные различного типа (уже преобработанная информация). На их основе производится классификация состояний элементов с помощью классификаторов. Классификаторы выдают вектора $S_{i\text{act}}$, характеризующие состояния элементов. На основе конкатенации векторов производится распознавание особой (нежелательной, опасной) ситуации. Если такая ситуация идентифицирована, то результат конкатенации векторов применяется для отбора схожей ситуации из базы знаний. Отобранная ситуация, выступает ключом для извлечения из базы состава решения и его вывода пользователям.

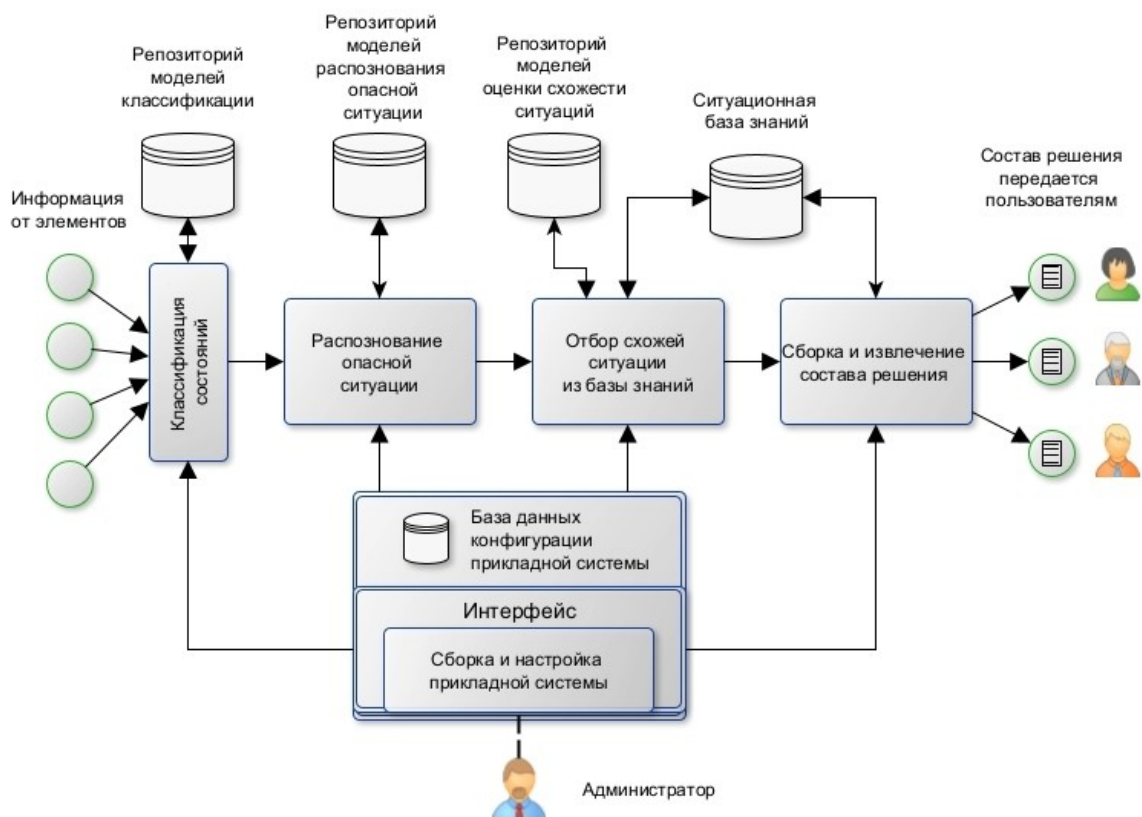


Рисунок 4.1 – Логическая архитектура Платформы

Создание предметно-ориентированной интеллектуальной системы с помощью средств Платформы реализуется через интерфейс администратором.

Дальнейшее функционирование системы ведется в режиме реального времени. Отмеченные пользователи – predeterminedенные адресаты для получения решения.

4.2 Физическая архитектура Платформы

Для обеспечения выполнения функций предусмотрены следующие основные модули:

- Ядро-оркестратор (включает подмодуль сбора комплексной ситуации; подмодуль идентификации опасной ситуации; подмодуль выдачи решения);
- Репозиторий моделей классификации состояний элементов (набор модулей для обучения и применения моделей машинного обучения и инженерии знаний);
- Репозиторий моделей распознавания опасной ситуации;
- Репозиторий моделей оценки схожести ситуаций;
- База данных конфигурации прикладной системы;
- База знаний, где хранятся прецеденты (ситуации и их решения);
- База данных конфигураций обученных классификаторов (включает базу правил логического вывода для классификаторов на экспертных правилах);
- База данных конфигураций обученных нейросетевых моделей классификаторов (включает базу правил логического вывода).

Физическая архитектура с составом основных модулей представлена на рисунке 4.2.

Модуль настройки (сборки) платформы реализован с помощью средств библиотеки Python Flask. Библиотека обеспечивает реализацию Платформы в виде web-приложения с интерфейсом, что позволяет создавать масштабируемую фреймворк-платформу, понятную для конечного пользователя, не имеющего опыта взаимодействия с кодом и нейросетевыми моделями.

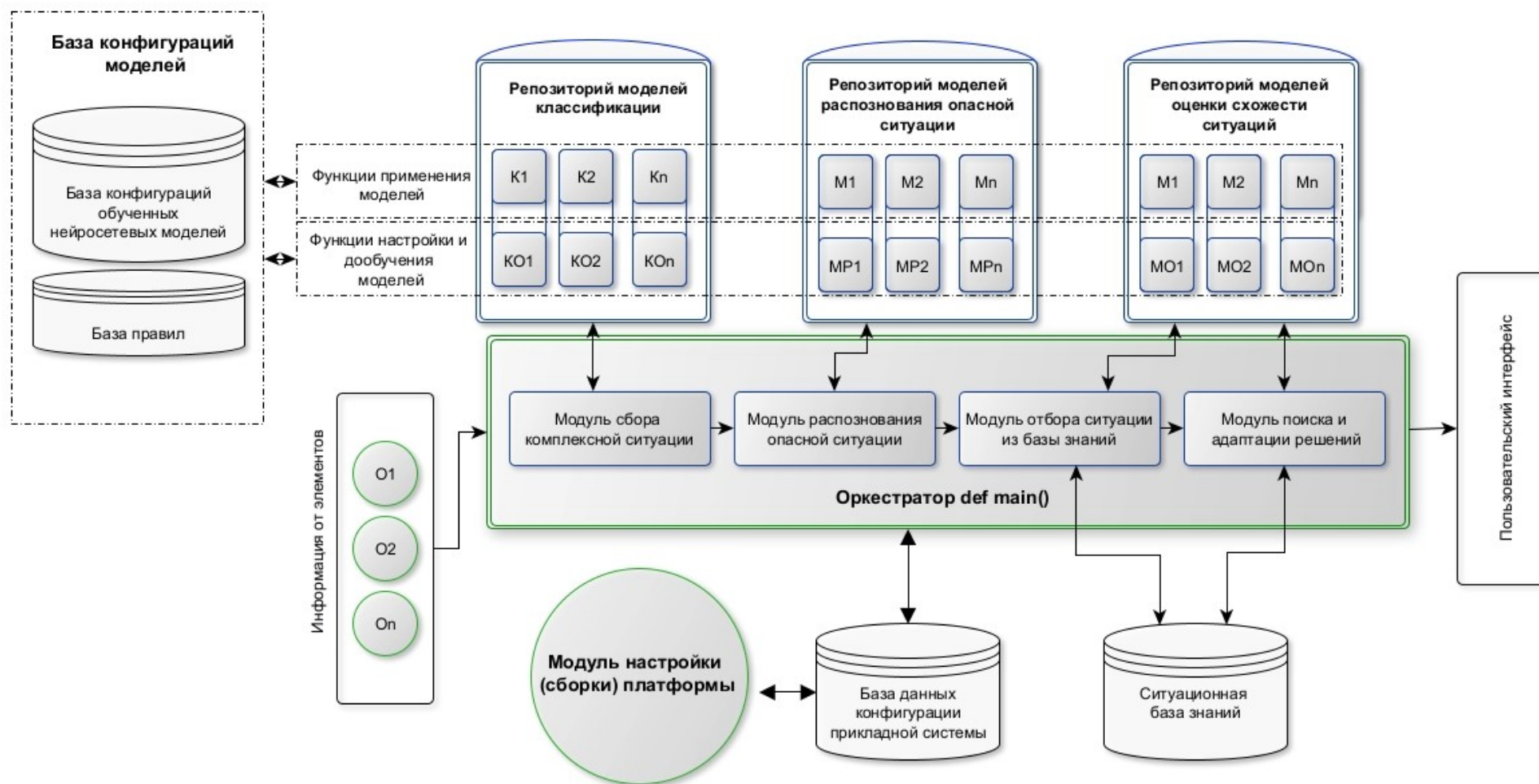


Рисунок 4.2 – Физическая архитектура

Модуль настройки включает в себя набор веб-страниц с интерфейсом, необходимым для создания предметно-ориентированной интеллектуальной системы вывода решений. Подробнее в п. 4.4

Идентификация состояний элементов производится в подключаемых модулях-классификаторах из репозитория классификаторов состояния на основе моделей машинного обучения и инженерии знаний. Классификаторы на вход получают данные различного типа. За каждым элементом закреплен персональный обученный классификатор, который с помощью нейросетей или системы логических правил интерпретирует входные сигналы в состояние. На выходе от каждого классификатора вектор $S_{i\text{act}}$, описывающий состояние элемента.

Работой настроенной запущенной системой руководит оркестратор (def main). Его функция заключается в реализации алгоритма гибридного CBR. Оркестратор с помощью набора модулей поочередно опрашивает элементы, запуская классификаторы, формирует вектор комплексной ситуации \check{S}_{act} , описывающий комплексную ситуацию на объекте. На его основе идентифицируется комплексная ситуация. В случае обнаружения опасной ситуации запускается в работу модуль отбора ситуации из базы знаний. Модуль отбора ситуации обращается к определенной модели из репозитория на основе данных конфигурации прикладной системы и получает от нее идентификатор ситуации. Далее модуль выдачи решения, получая на вход идентификатор ситуации, обращается к базе знаний и выдает пользователям состав решения. Более подробно архитектура оркестратора рассмотрена в разделе 4.3.

База данных, база знаний реализованы с помощью средств SQLite и файлов структуры .json (для ряда задач модулей на основе экспертных правил), являются неотъемлемыми частями системы, обеспечивают следующие функции: ведение базы правил логического вывода, ведение базы прецедентов в виде пар «Ситуация-Решение», сборка предметно-ориентированной системы.

База данных хранит в себе информацию о внесенных элементах, необходимую для обучения и функционирования классификаторов, информацию о настроенном модуле отбора ситуации.

База знаний содержит вектора \check{S}_z и составы решений. Подробнее рассмотрена в разделе 4.5.

Базы конфигураций обученных моделей для оценки схожести и классификации представляют собой директории с файлами конфигураций и весов обученных нейросетей, а также таблицы с экспертными правилами (для моделей на правилах логического вывода).

4.4 Алгоритм сборки предметно-ориентированной системы

Общие шаги по сборке предметно-ориентированной системы подразумевают подключение к ядру и настройку моделей из репозитория согласно доступным данным, что подразумевает выполнение следующих действий:

1. Регистрация элементов сложного технологического объекта;
2. Выбор и назначение классификатора для каждого элемента;
3. Обучение и настройка классификаторов;
4. Выбор модели оценки схожести ситуации;
5. Настройка модели оценки схожести ситуации;
6. Мониторинг актуальных состояний элементов;
7. Получение состава решения.

4.5 Базовый набор модулей

Оркестратор

Модуль оркестратор (функция `def main()`) разработан с учетом возможности масштабирования разрабатываемой на основе Платформы прикладной системы.

Архитектура модуля представлена в Приложении 2.

Архитектура модуля позволяет дополнять Платформу пользовательскими моделями в репозитории в виде самостоятельных модулей.

Такие возможности обеспечивает подход объектно-ориентированного программирования и использования кодовых названий моделей классификаторов и сравнения ситуаций.

Модели хранятся в различных модулях (файлах расширения .py) и для применения модели для конкретного элемента применяется типовая функция этой модели `<anyclassifier>.apply(id)`. Функция срабатывает с применением персональной для элемента конфигурации модели, созданной в результате обучения классификатора на этапе сборки прикладной системы. Конфигурация модели может содержать веса и конфигурацию обученной нейросети, заданные экспертные правила, имена состояний и другое.

Соответственно, функция каждой модели универсальна, но благодаря конфигурациям она срабатывает по-разному и выдает свой ожидаемый результат для каждого элемента.

Каждая модель имеет кодовое название, например, «nnwtable». Кодовое название автоматически заносится в БД конфигурации на этапе настройки системы при регистрации элементов и хранится там в формате переменной string.

Обращаться к моделям по кодовым названиям позволяет класс `modules` и функция `def callMethod(f, a, id)`, где `f = modules()` (ссылка на класс, содержащий адрес модели), `a` – кодовое название классификатора в виде переменной `str`, `id` – номер элемента.

Каждой модели в классе `modules` соответствует атрибут с кодовым названием, например, то же «nnwtable». Атрибут содержит: функцию вызова модели, команду добавления вектора $S_{i \text{ act}}$ в общий список.

`def callMethod(f, a, id)`, получая на вход `id` элемента, название модели, понимает какую именно нужно запустить модель, обращается к атрибуту класса `modules` и запускает его, при этом функции модели передается `id` элемента, по которому можно подгрузить нужную конфигурацию из репозитория, относящуюся именно к этому элементу.

Работа оркестратора состоит из нескольких последовательных этапов.

1) Обращение к базе данных конфигурации прикладной системы и получение всех номеров (id) зарегистрированных элементов и их классификаторов в виде строчной переменной (кодированное название классификатора – атрибут БД «adresseduc»).

2) Поочередный запуск классификаторов для каждого элемента.

После выполнения функции классификации возвращается вектор $S_{i \text{ акт}}$, описывающий состояние элемента, затем добавляется в общий список векторов от всех элементов.

3) Проверка состояния элемента на допустимость в соответствии с указанными в конфигурации правилами.

4) После выполнения работы классификаторов всех элементов, образуется вектор комплексной ситуации $\check{S}_{\text{акт}}$. Происходит проверка комплексной ситуации на допустимость

5) Если идентифицирована опасная ситуация, запускается назначенная модель сравнения ситуации. Возвращается идентификатор отобранной ситуации.

6) Происходит выдача состава решения.

Фрагмент кода представлен ниже:

```
class modules:
    def nnwtable(self, id): #<-- атрибут class modules с именем
        «nnwtable» - кодированное название классификатора, которое
        записывается и хранится в БД конфигурации
        a = nnwtable.nnwtableapply(id) #<--вызов функции
        классификатора, возвращается вектор состояния элемента
        itog.append(a) #<--добавление вектора состояния
        элемента к общему вектору (конкатенирование)
        check(id, a) #<--проверка условий допустимости
        состояния
        return

    def nnwimages(self, id): #<-- атрибут class modules с
        именем «nnwimages» - кодированное название классификатора, которое
        записывается и хранится в БД конфигурации
        id = id
        a = yolo.nnwimages(id) #<--вызов функции
        классификатора, возвращается вектор состояния элемента
```

```

        itog.append(a)      #<--добавление вектора состояния
элеента к общему вектору (конкатенирование)
        check(id, a)
        return

    def callMethod(f, name, id):
        getattr(f, name)(id) # вызов атрибута класса (функции) по
кодovому названию классификатора
    f = modules()
    def compile():
        all = Elements.query.all()# обращение к БД конфигурации
        for i in all: # цикл по каждому элементу
            id = i.id # получение id элемента
            name = i.adresseduc # получение кодового названия
классификатора, назначенного элементу
            callMethod(f, name, id) # вызов функции обращения к
классу

```

Оркестратор перезапускается через определенное настроенное время (определяется экспертным путем в зависимости от особенностей функционирования объекта и его элементов). Весь алгоритм работает в цикле.

Модуль настройки (сборки) платформы

Модуль настройки реализован с помощью библиотеки Flask. Представляет собой набор функций, связанных с веб-страницами с интерфейсом, классы таблиц базы данных конфигурации системы, назначенные директории служебных файлов.

Для страниц разработаны html, css файлы.

Модуль настройки предназначен для взаимодействия администратора и пользователя с Платформой на этапе создания прикладной системы и ее эксплуатации.

Позволяет выполнять функцию: обеспечение создания предметно-ориентированной интеллектуальной системы вывода решений на прецедентах в ситуациях, возникающих на сложных технологических объектах.

Предусмотрена функция регистрации элемента. Помимо регистрации предусмотрена функция удаления элемента, которая убирает все записи из БД

конфигурации о элементе и ликвидирует директорию конфигурации обученного классификатора (при наличии).

Функция запуска обучения классификатора позволяет пользователю загрузить через интерфейс обучающий набор данных. После проверки файла, функция передает обучающий файл самой функции обучения, находящейся в репозитории моделей классификаторов.

База данных конфигурации прикладной системы

База данных конфигурации состоит из двух таблиц и содержит информацию, необходимую для работы созданной прикладной системы. Таблицы заполняются на этапе создания системы с помощью интерфейса Платформы и могут дополняться в процессе эксплуатации.

Логическая модель данных БД конфигурации представлена на рисунке 4.3.

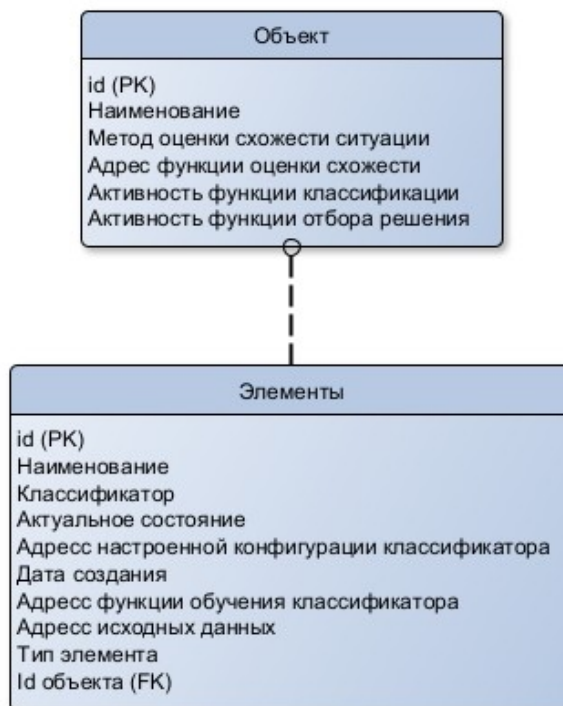


Рисунок 4.3 – Логическая модель данных БД конфигурации системы

БД реализована с помощью SQLite. Взаимодействие осуществляется с помощью модуля настройки системы через модуль SQLAlchemy библиотеки flask_sqlalchemy.

База знаний системы

База знаний состоит из трех таблиц (обозначения согласно разделам 2.3, 2.4):

- таблица с векторами \check{S}_z , которые выступают в роли ключа для отбора ситуации;
- таблица мультивекторов R^y_z (формальное представление решения Sol некоторой ситуации Sit), связанных с ключом \check{S}_z ;
- таблица компонентов решения, соответствующих каждому шагу из последовательности R^y_z .

Логическая модель данных БЗ представлена на рисунке 4.4.

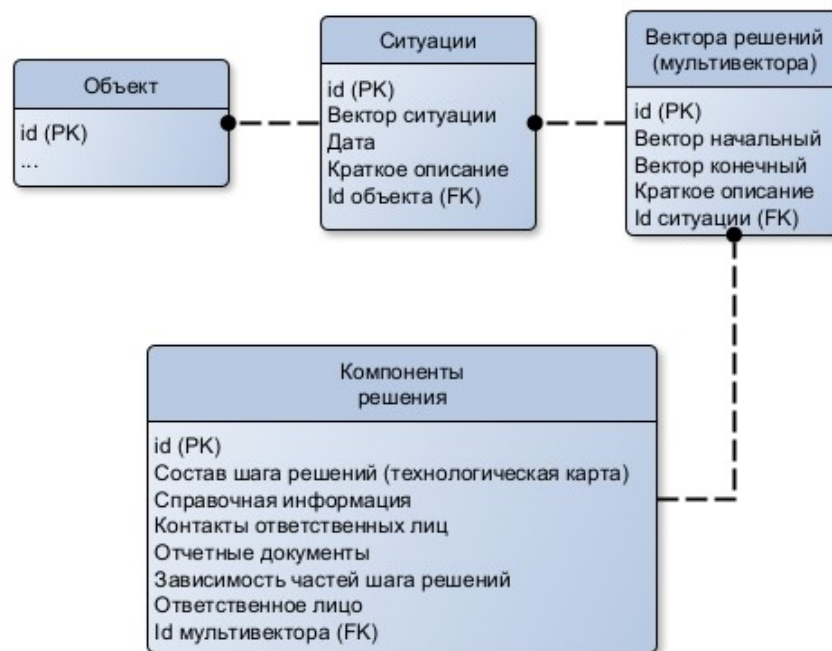


Рисунок 4.4 – Логическая модель данных базы знаний

БД реализована с помощью SQLite. Взаимодействие осуществляется через модуль SQLAlchemy библиотеки flask_sqlalchemy.

Содержание репозитория классификаторов и моделей оценки схожести

Исходя из гетерогенности потенциальных входных данных разработано 8 модулей классификации состояний с учетом особенности входных данных.

Каждый модуль включает в себя функции обучения/настройки классификатора, функцию-обработчик ошибок и саму модель классификатора.

Набор модулей классификации обеспечивает функции: классификация состояний элементов сложного объекта по входным данным; заполнение и ведение базы правил логического вывода (для классификаторов на основе экспертных правил). Модели разработаны для различных типов данных: табличные качественные и количественные, логические, изображение.

В Платформу включено 2 модуля для оценки схожести ситуаций: модель на основе аддитивной свертки, обучаемая модель выбора ситуации, разработанная на основе математической модели, представленной в разделе 3.1. Каждый модуль включает в себя функции обучения/настройки модели сравнения, функцию-обработчик ошибок и саму модель оценки схожести.

Набор моделей сравнения ситуации обеспечивает функции: вывод решения по входным данным о ситуации. Модели разработаны для работы с ситуациям, формализованными в виде вектора \check{S} (см. главу 2).

Модель сравнения ситуации получает на вход следующие данные: вектор актуальной ситуации \check{S}_{act} , вектора ситуаций из базы знаний \check{S}_z , переводит входные данные в вычисляемое значение схожести $f(\check{S}_{act} \cdot \check{S}_z) = Sim$, определяет идентификатор пяти наиболее схожих ситуаций из базы знаний, выдает идентификатор модулю выдачи решения.

4.6 Алгоритм генерации ситуаций (обучающих множеств)

Задача аугментации ситуаций-примеров.

Определение схожести объектов с помощью моделей машинного обучения требует значительного объема обучающих данных. В контексте задачи определения схожести ситуаций, заданных векторами, обучающие данные заданы парами ситуаций и меткой их схожести. Метка может быть количественная оценка схожести или в упрощенном виде 0/1, обозначающая «схож»/ «несхож» [75, 76].

На практике возможны случаи недостатка обучающих данных. Формирование наборов данных становится самостоятельной и актуальной научно-практической задачей. Ее решение в условиях недостатка обучающих данных и при отсутствии возможностей их сформировать по наблюдениям за системой, связывают с идеей расширения (аугментации) того небольшого числа обучающих примеров, которые уже имеются [117].

Другое практическое применение заключается в пополнение базы прецедентов. Реализуется алгоритм генерации новой ситуации и под нее адаптируется имеющееся решение из базы знаний (см. раздел 3.2).

Для генерации ситуаций (множества ситуаций) предлагается использовать метод преобразований для создания обучающего набора данных на основе примеров. Метод состоит в следующем. Пусть $SIT = \{Sit_r | r = 1, \dots, R\}$ таких, что любые две ситуации имеют $Sim(Sit_r, Sit_p) < Th$, где Th – порог, после превышения которого можно говорить о схожести ситуаций.

Применительно к базе знаний в системе с выводом на прецедентах это означает, что есть <Ситуация (Sit_r), Решение (Sol_r)> в количестве R , где все решения отличаются друг от друга. Иначе говоря, признак решения выступает в качестве классификационного признака, разделяющего все множество ситуаций на классы.

Таким образом, задача состоит в том, чтобы путем преобразований расширить имеющееся множество SIT новыми элементами так, чтобы для каждого R получить множество $SIT_r = \{Sit_r, Sit_1, Sit_2, \dots\}$, где все элементы – ситуации «схожи» с Sit_r в соответствии с заданным критерием схожести.

Алгоритмы решения задачи преобразований. В основе предлагаемых алгоритмов – метод покоординатного спуска. Исходная точка поиска – вектор ситуации в пространстве состояний [118]. Первый алгоритм (рисунок 4.5) является базовым, он позволяет получить на выходе множество ситуаций, удовлетворяющих требованиям схожести с исходным примером Sit .

```

1. Начало
2. Устанавливаем  $Sit_{out} = Sit$ 
3. Для  $j$  от 1 до  $M$ , делать
//  $M$  - число компонентов вектора ситуации в пространстве состояний
{
4. Преобразуем значение  $j$ -го компонента вектора ситуации на противоположное
 $Sit_{out}[j] = |Sit[j] - 1|$ 
5. Вычисляем  $h(Sit_{out})$ .
6. Если  $h(Sit_{out}) > Th$  То
//Включаем  $Sit_{out}$  в искомое множество  $SIT$ , если новая ситуация удовлетворяет порогу сходства и значение  $h(Sit_{out})$  в множество меток  $SIM$ 
7.  $Sit_{out}$  in  $SIT$  и  $h(Sit_{out})$  in  $SIM$ 
}
8. Упорядочиваем элементы множества  $SIT$  по убыванию значения  $h(Sit_{out})$ 
9. Отбираем из  $SIT$  первые  $V$  элементов (где  $V$  заданное число искомых элементов) и
10. Добавляется начальное преобразуемое  $Sit$ , формируем выходное множество  $SIT'$  - расширенное множество для ситуации  $Sit$  и, соответственно, множество меток  $SIM'$ .
11. Конец

```

Рисунок 4.5 – Алгоритм 1

Повторение этого алгоритма по всем r позволяет из множества $SIT = \{Sit_r | r = 1, \dots, R\}$ получить расширенные множества исходных ситуаций-примеров $\{SIT'_r | r = 1, \dots, R\}$ и их меток:

$$SIT'_r = \{ Sit'_k | k = 1, \dots, R_r \},$$

$$SIM'_r = \{ Sim_k | k = 1, \dots, R_r \},$$

где для всех k имеем $Sim(Sit_1, Sit_k) > Th$, т.е. все элементы множества ситуаций удовлетворяют требованию сходства с исходной ситуацией (исходная ситуация включена в множество в виде Sit_1).

Рассмотрим комплексную ситуацию на сложном технологическом объекте. Задача (4.1) может быть записана как многокритериальная оптимизационная задача. Задача в таком случае звучит следующим образом – необходимо найти такую комплексную ситуацию, что:

$$H(\text{Sit}') = \text{Agr}(\text{Sim}_1, \dots, \text{Sim}_i, \dots, \text{Sim}_N) \rightarrow \max \quad (4.2)$$

при ограничениях $\text{Sim}_k \geq \text{Th}_k$ при $k \in K$,

где K – множество индексов тех элементов сложного объекта, по которым обязательно должно быть достигнуто сходство не ниже некоторого порога Th_k .

Здесь Sim_i – функция сходства между ситуациями по i -му элементу $\text{Sim}_i = \text{Sim}(\text{Sit}_i, \text{Sit}'_i)$.

Алгоритм 2 формирования вектора комплексной ситуации на базе алгоритма 1 представлен на рисунке 4.6.

```

1. Начало
2. Для  $i$  от 1 до  $N$  делать
   {
3. Выполнить Алгоритм 1.
   //На выходе Алгоритма 1 формируется множество множеств  $\text{SIT}'_1, \text{SIT}'_2, \dots, \text{SIT}'_i, \dots, \text{SIT}'_N$ ,
   где  $\text{SIT}'_i = \{ \text{Sit}'_{ik} \mid k = 1, \dots, R_i \}$ ,  $R_i$  – число векторов
   ситуаций, сгенерированных путем преобразований исходной ситуации и
   удовлетворяющих критерию схожести (4.1).
   }
   //Устанавливаем начальные значения компонентов мультивектора
   выходной ситуации, в который включаются первые элементы каждого из
   множеств
4.  $\text{Sit}_{\text{out}} = (\text{Sit}'_{i1} \mid i = 1, \dots, N)$ 
5. Для  $i$  от 1 до  $N$  делать
6.     Для  $k$  от 1 до  $R_i$  делать
       {
7.          $\text{Sit}_{\text{out}} = (\text{Sit}'_{ik})$ 
8.          $H = H(\text{Sit}_{\text{out}})$ 
9.         Если  $H > \text{MAXH}$  То
           {
10.              $\text{MAXH} = H$ 
11.              $\text{Sit}' = \text{Sit}_{\text{out}}$ 
           }
       }
12. Конец.
```

Рисунок 4.6 – Алгоритм 2

Для учета контекста и окружения, которые могут влиять на решение, но не поддаются управлению, задается значение $\text{Th}_k = 1$. Для решения задачи (4.2) используется Алгоритм 2, где на вход подается конкатенация векторов состояний.

В этом случае число M заменяется на $M \cdot N$ (количество из N векторов по M компонентом), а вместо локального сходства $h(.)$ определяется глобальное сходство $H(.)$.

На выходе алгоритма формируется один мультивектор ситуации Sit' , компоненты которого – вектора из множеств $SIT'_1, SIT'_2, \dots, SIT'_i, \dots, SIT'_N$ и оценка схожести $H(Sit')$ удовлетворяет критерию (4.2).

Подробнее алгоритмы и экспериментальная часть описаны в работе [119].

4.7 Разработка тестового набора прецедентов для ситуационной базы знаний

Тестовый набор прецедентов был разработан для объекта «Котельная жилого комплекса здания». Данный объект имеет в составе 18 элементов (таблица 4.2).

Таблица 4.1 – Состав сложного объекта «Котельная жилого комплекса здания»

Элементы	Возможные состояния	Вектор S
Ввод холодной воды	Нормальное давление/	[1]
	Опасное давление	[0]
Гидроаккумулятор №1	Исправно/	[1]
	Низкое давление	[0]
Гидроаккумулятор №2	Исправно/	[1]
	Низкое давление	[0]
Насос №1	Работает/	[1,0,0,0]
	Набор мощности/	[0,1,0,0]
	Неисправность/	[0,0,1,0]
	Выключен	[0,0,0,1]
Насос №2	Работает/	[1,0,0,0]
	Набор мощности/	[0,1,0,0]
	Неисправность/	[0,0,1,0]
	Выключен	[0,0,0,1]

Продолжение таблицы 4.1

Аварийный электрокотел №1	Работает/ Набор мощности/ Неисправность/ Выключен	[1,0,0,0] [0,1,0,0] [0,0,1,0] [0,0,0,1]
Котел №1	Работает/ Неисправность/ Авария Выключен/ Отсутствие газа на входе/ Не доступен	[1,0,0,0,0,0] [0,1,0,0,0,0] [0,0,1,0,0,0] [0,0,0,1,0,0] [0,0,0,0,1,0] [0,0,0,0,0,1]
Котел №2	Работает/ Неисправность/ Авария Выключен/ Отсутствие газа на входе/ Не доступен	[1,0,0,0,0,0] [0,1,0,0,0,0] [0,0,1,0,0,0] [0,0,0,1,0,0] [0,0,0,0,1,0] [0,0,0,0,0,1]
Система газозулавливания	Обнаружена загазованность/ Газ не обнаружен/ Система недоступна	[1,0,0] [0,1,0] [0,0,1]
Электропитание	Включено/ Выключено	[1] [0]
Аварийная группа	Доступно/ Недоступно	[1] [0]
Сантехник	Доступен/ Недоступен	[1] [0]
Электрик	Доступен/ Недоступен	[1] [0]
Вход в помещение	Свободно/ обнаружено препятствие	[1] [0]
Подъезд к территории	Свободно/ обнаружено препятствие (ворота перегорожены транспортом)	[1] [0]

Продолжение таблицы 4.1

Территория	Обнаружен транспорт/	[1,0,0,0]
	Обнаружены люди/	[0,1,0,0]
	Обнаружены люди и транспорт/	[0,0,1,0]
	Пусто	[0,0,0,1]
Природные явления	Дождь сильный/	[1,0,0,0]
	Снег/	[0,1,0,0]
	Гололед/	[0,0,1,0]
	Не влияют	[0,0,0,1]
Помещение	Нормальное состояние/	[1]
	Обнаружено изменение	[0]
Температура воздуха на улице	Тепло/	[1]
	Холодно (ниже -10°C)	[0]

Ситуация на объекте описывается вектором \check{S} размерностью 46 позиции, в случае исправности и отсутствия влияния вектор будет иметь вид:

$$[1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,1,1,1,1,1,0,0,0,1,0,0,0,1,1,1].$$

Тестовый набор включает 12 прецедентов (ситуация и решение). Пример прецедента из набора представлен ниже.

Прецедент. Обнаружена неисправность на объекте: система газоулавливания обнаружила газ в помещении, на газовые котлы не поступает газ. Контекст характеризуется следующим: препятствие перед въездом (обнаружен транспорт у ворот), температура на улице – холодно, штатный электрик – недоступен.

Программа действий содержит 5 шагов по устранению неисправности и возобновлению работы котлов с учетом контекста.

Вектор ситуации, выступающий в роли ключа для отбора, имеет следующий вид:

$$\check{S}^1 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,0,0,0,1,1,0].$$

Вектор конечной ситуации имеет следующий вид:

$$\check{S}^6 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0].$$

Решение для прецедента №6.

Расшифровка начального вектора \check{S}^1

$$\check{S}^1 = [1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0]$$

Таблица 4.2 – Расшифровка начального вектора \check{S}^1 с указанными типами входных воздействий для каждого элемента

Элементы	Состояние	Тип входных воздействий	Вектор S
Ввод холодной воды	Нормальное давление	Параметры технических элементов	[1]
Гидроаккумулятор №1	Исправно	Параметры технических элементов	[1]
Гидроаккумулятор №2	Исправно	Параметры технических элементов	[1]
Насос №1	Работает	Параметры технических элементов	[1,0,0,0]
Насос №2	Работает	Параметры технических элементов	[1,0,0,0]
Аварийный электродкотел №1	Работает	Параметры технических элементов	[1,0,0,0]
Котел №1	Отсутствие газа на входе	Информация о наличии/отсутствии потока обеспечивающего ресурса	[0,0,0,0,1,0]
Котел №2	Отсутствие газа на входе	Информация о наличии/отсутствии потока обеспечивающего ресурса	[0,0,0,0,1,0]
Система газоподогрева	Обнаружена загазованность	Параметры технических элементов	[1,0,0]

Продолжение таблицы 4.2

Электропитание	Включено	Информация о наличии/отсутствии потока обеспечивающего ресурса	[1]
Аварийная группа	Доступно	Информация о доступности/недоступности персонала	[1]
Сантехник	Доступен	Информация о доступности/недоступности персонала	[1]
Электрик	Недоступен	Информация о доступности/недоступности персонала	[0]
Вход в помещение	Обнаружен о препятствие	Информация о параметрах окружения	[0]
Подъезд к территории	Свободно	Информация о параметрах окружения	[1]
Территория	Пусто	Информация о параметрах окружения	[0,0,0,1]
Природные явления	Не влияют	Информация о параметрах окружения	[0,0,0,1]
Помещение	Нормальное состояние	Информация о параметрах окружения	[1]
Температура воздуха на улице	Холодно (ниже - 10°C)	Информация о параметрах окружения	[0]

Состав решения

Программа действий, примененных в ситуации, имеет следующие шаги:

1) Краткое описание: Немедленная реакция: отключение электроэнергии – снижение рисков возгорания газа, вызов газовой службы.

Вектор начальный:

$$\check{S}^1 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,1,1,1,0,0,1,0,0,0,1,0,0,0,1,1,0]$$

Вектор конечный:

$$\check{S}^2 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,1,1,0]$$

Состав решения:

Ответственные: диспетчер.

1. Диспетчер принимает заявку и проводит инструктаж с заявителем о мерах безопасности до прибытия аварийной бригады.

2. Аварийная заявка регистрируется, и ее содержание оформляется для передачи аварийной бригаде. Это может быть сделано либо через документацию, либо по радиотелефонной связи.

Ответственные: руководитель аварийной бригады.

1. Руководитель аварийной бригады проводит краткий инструктаж по выполнению газоопасных работ на объекте аварии, а также разъясняет меры безопасности и подготавливает необходимую исполнительную документацию.

2. Аварийная бригада должна выехать на место происшествия не позднее чем через 5 минут после получения заявки.

3. На въездах к аварийному объекту устанавливаются предупредительные знаки, а также принимаются меры для предотвращения возникновения открытого огня и ограничения доступа людей в загазованные помещения.

Ответственные: дежурная аварийная группа жилого комплекса.

1. Отключение питания помещения котельной (секция котлов) в диэлектрических перчатках: Рубильник ВНК-37-1 ЗПН, 3 секция, ВРУ 400А (Адрес: ул. Тракторная 158/1 – напротив центральных ворот въезда через дорогу). Вывесить запрещающий плакат.

2. Перекрытие запорной арматуры на входе с целью локализации аварии на поврежденном участке газопровода.

3. Дождаться прибытия группы оперативного реагирования газовой службы.

4. Постараться вызвать владельцев транспорта перед въездом.

2) Краткое описание: устранение препятствий на въезде.

Вектор начальный:

$$\check{S}^2 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,1,1,0]$$

Вектор конечный:

$$\check{S}^3 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

Ответственные: дежурная аварийная группа жилого комплекса.

1. Если хозяина не удалось дождаться необходимо вызвать службу экстренной эвакуации.

Контактные данные: +79055555555.

Газовая служба может отодвинуть транспорт своими силами при надобности.

3) Краткое описание: Поиск утечки газа, устранение утечки.

Ответственные: руководитель аварийной бригады.

Вектор начальный:

$$\check{S}^3 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

Вектор конечный:

$$\check{S}^4 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

1. Определение концентрации газа в помещении котельной с использованием приборного метода.

2. Проведение интенсивной вентиляции котельной и организация постоянного контроля за уровнем газа.

3. Поиск места утечки газа с помощью приборного метода или пенообразующего раствора на газопроводе и газоиспользующем оборудовании котельной.

4. Выполнение работ по ликвидации аварийной ситуации, включая устранение нарушений герметичности соединений на входе газопровода в помещение.

5. Составление акта об аварийно-диспетчерском обслуживании сети газопотребления.

Ответственные: аварийная бригада ЖК

1. Запуск электродвигателя по разрешению аварийной газовой бригады на время ремонтных работ

4) Краткое описание: Восстановление подачи электроэнергии в секцию газовых котлов.

Вектор начальный:

$$\check{S}^4 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

Вектор конечный:

$$\check{S}^5 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,1,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

Ответственные: аварийная служба

1. Снять запрещающий плакат. Включить питания помещения котельной (секция газовых котлов) в диэлектрических перчатках: Рубильник ВНК-37-1 ЗПН, 3 секция, ВРУ 400А (Адрес: ул. Тракторная 158/1 – напротив центральных ворот въезда через дорогу).

5) Краткое описание: восстановление подачи газа и перезапуск оборудования.

Вектор начальный:

$$\check{S}^5 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,1,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

Вектор конечный:

$$\check{S}^6 = [1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,1,1,1,0,1,1,0,0,0,1,0,0,0,1,1,0]$$

Ответственные: оперативная бригада газовой службы.

1. Открыть запорную арматуру на вход

2. Приборным методом убедиться в допустимости концентрации газа в помещении в котельной.

Ответственные: аварийная бригада ЖК

1. Проверьте, чтобы не было утечек газа.
2. Выключите главный выключатель (ВКЛ / ВЫКЛ).
3. Если у вас установлен комнатный термостат, настройте его на нужную температуру для активации нагрева.
4. Убедитесь в правильном давлении газа и дайте котлу прогреться в течение нескольких минут.
5. Настройте горелку в соответствии с местными стандартами и нормами.
6. Установите желаемую температуру нагрева теплоносителя через панель управления котлом.
7. После 5 минут работы выпустите воздух из отопительного контура и восстановите давление до 1,5 бар.
8. При необходимости снова удалите воздух из системы отопления и заполните ее водой для достижения нужного давления.
9. Убедитесь, что отопительная система сбалансирована, и при необходимости отрегулируйте клапаны, чтобы избежать нарушений в циркуляции теплоносителя через котел.

Последующая настройка

1. Проверьте систему на наличие утечек.
2. Убедитесь, что скорость потока теплоносителя через котел достаточна, следуя этим шагам:
 3. Убедитесь, что котел работает на максимальной мощности.
 4. После стабилизации температуры теплоносителя зафиксируйте значения на подаче и возврате в котел.
 5. Проверьте, чтобы разница между этими значениями не превышала 20К.
 6. Если разница превышает 20К, проверьте настройки насоса и правильность установки системы.

4.8 Задача оценки схожести ситуаций. Экспериментальная работа

Описание данных.

На основе сложного технологического объекта — теплового пункта здания — был проведен ряд экспериментов, направленных на выбор ситуации из базы знаний. Основная цель этих экспериментов заключалась в определении наиболее эффективного инструмента для задач выбора ситуаций по критерию схожести.

Тепловой пункт здания соединяет систему централизованного теплоснабжения района или города с системой отопления дома. Технологическая схема представляет собой независимую двухконтурную систему отопления, в которой внешний теплоноситель передает тепловую энергию через теплообменник теплоносителю системы отопления.

В качестве компонентов сложного объекта были рассмотрены следующие разнородные элементы, сгруппированные по категориям:

Оборудование: узел ввода технической воды, теплообменник, насос, внутренняя труба, программное обеспечение, электричество, релейная защита.

Операционное окружение: оборудование энергообеспечения, службы сервиса, аварийно-ремонтные бригады.

Контекст: помещение теплового пункта, соседние здания, природные объекты и явления.

Элементы группы «контекст» не связаны напрямую с объектом, но считаются его частью, так как могут оказывать влияние на него. Например, снегопад может затруднить доступ персонала к объекту и повлиять на принятие решений в проблемной ситуации.

Множество состояний, в которых могут находиться эти элементы, содержит 8 типовых состояний («исправен» или «неисправен», «доступен» или «недоступен» и т.п.).

На основе описанного объекта были подготовлены наборы данных: обучающие и валидационные. Обучающий набор данных необходим для обучения нейросети, т.е. определения весов связей между нейронами [49].

Валидационный набор используется для оценки производительности обученной нейросети. Для этого обученная нейросеть прогнозирует классификации примеров в валидационном наборе. Эти предсказания сравниваются с истинными классификациями примеров, чтобы оценить точность модели [50].

Для проведения экспериментов было подготовлено шесть наборов данных, содержащих пары ситуаций на объекте «тепловой пункт здания» с известными значениями Sim, по три набора для четкой и нечеткой классификации:

1Т. 100 пар для обучения. При этом значения Sim подобраны так, что 50% из пар ситуаций включают в себя похожие ($Sim > 0,8$), а 50% - непохожие ситуации.

2Т. 100 пар для обучения. Включают в себя 5 наборов по 20 пар, где одна ситуация сравнивается с другими двадцатью. В каждом наборе значения Sim пар ситуаций равномерно распределены в диапазоне $[0,1]$. При этом присутствуют 6-8 пар со значением $Sim > 0,75$, ранжированные экспертом по возрастанию Sim.

3V. 60 пар для валидации. Включают в себя 3 набора по 20 пар (3V.1, 3V.2, 3V.3), где одна ситуация сравнивается с другими двадцатью. Наборы подаются по отдельности. При этом присутствуют 6 пар со значением $Sim > 0,75$, ранжированные экспертом по возрастанию Sim.

Исследование моделей для определения схожести ситуаций

В исследовании был проведен эксперимент по анализу производительности пяти нейросетевых моделей для оценки схожести ситуаций и их комбинаций. Каждая модель оперирует входными данными, получаемыми из мультивекторов двух сравниваемых ситуаций.

Базовой архитектурой для моделей выбрана известная архитектура multi-layer network (MLN) с одним выходным нейроном и входным слоем X_{in} , число нейронов на котором определяется длиной входного вектора. Базовая архитектура слоев представлена на рисунке 4.7, встроенные функции активации ReLU и Swish библиотеки Keras чередуются.

На основе базовой архитектуры реализованы три модели M, которые отличаются числом входных элементов (векторами X_{in}), числом элементов на

промежуточных слоях и количеством промежуточных слоев, подбираемых экспериментально:

Модель M1 получает:

$$X_{in} = \text{concat}(\check{S}_{act}, \check{S}_z),$$

$$M1: X_{in} \rightarrow \text{Sim} \in [0,1].$$

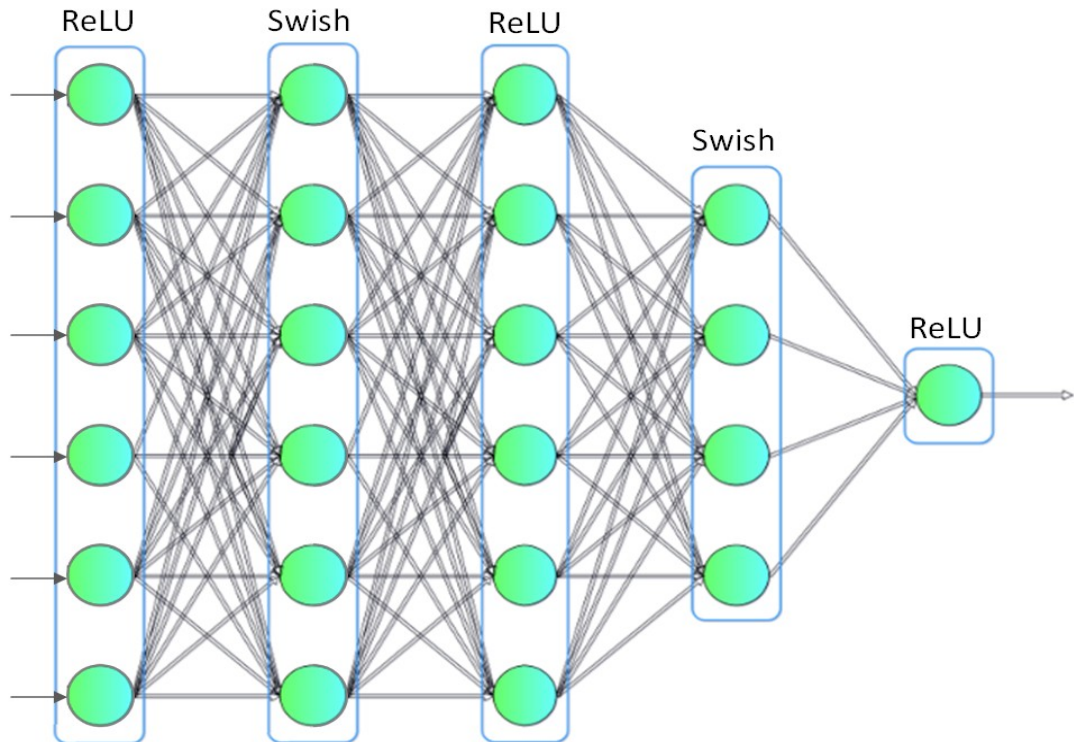


Рисунок 4.7 – Архитектура нейросетевой модели MLN

Модель M2a получает посимвольную разницу мультивекторов двух ситуаций:

$$X_{in} = |\check{S}_{act} - \check{S}_z|,$$

$$M2a: X_{in} \rightarrow \text{Sim} \in [0,1].$$

Модель M2b, которая на вход получает разницу мультивекторов, вычитаемую из 1, и на выходе выдает расстояние Dist между ними.

$$X_{in} = 1 - |\check{S}_{act} - \check{S}_z|,$$

$$M2b: X_{in} \rightarrow \text{Dist} \in [0,1].$$

Эти способы позволяют сократить длину входного вектора в два раза по сравнению с тем, который используется в модели M1. Сокращение размерности

входного вектора потенциально обладает положительным свойством, т.к. уменьшает количество комбинаций входных векторов, которые могут давать на выходе один и тот же результат вычислений.

Модель М3, которая на вход получает сумму:

$$X_{in} = |\check{S}_{act} + \check{S}_z|,$$

$$M3: X_{in} \rightarrow Sim \in [0,1].$$

Здесь компоненты входного для модели вектора X_{in} принимают значения от 0 до 2.

За основу следующей модели легла концепция сиамских нейронных сетей.

Модель М4: компаратор-сумматор.

Здесь отдельно сравнивались векторы $(\check{S}_{act} + \check{S}_z)$, их конкатенация подавалась на вход отдельного i -го компаратора. Компаратор имеет архитектуру многослойной полносвязной нейронной сети, которая определяет сходство между этими векторами. Каждый компаратор дает прогноз относительно сходства элементов в двух сравниваемых ситуациях.

Выходы компараторов в виде округленных значений функций активации Sigmoid 0 или 1 поступают на вход сумматора, назначением которого является вычисление итоговой оценки Sim.

Параметры каждого из компонентов архитектуры подбирались экспериментальным путем. В экспериментальных исследованиях пригодность этих моделей проверялась на валидационных наборах данных с помощью нескольких метрик качества, которые представлены далее.

Для оценки нейросетевых моделей использовались следующие метрики: MAPE (Mean Absolute Percentage Error), nDCG@k (Normalized Discounted Commulative Gain для лучших k), Accuracy.

Результаты исследования моделей

Каждая модель обучалась отдельно на наборах 1Т и 2Т, после чего оценивалось качество обученной модели на наборах 3V. Цикл обучения, валидации и анализа показателей производился 6 раз на одних и тех же данных для каждой модели в целях оценки стабильности показателей.

Объединение наборов 1Т и 2Т (60 пар набора 1Т и 80 пар набора 2Т) в набор 3Т позволило увеличить качество обученных моделей, что выразилось в снижении значения MAPE в среднем на 3%, верность ранжирования в среднем на 7%. По всей видимости, комбинирование наборов в один позволило охватить больший диапазон возможных ситуаций и одновременно указать на тенденцию выбора и ранжирования, которые присутствуют в наборе 2Т.

Далее проводился эксперимент по ансамблированию моделей (полученные результаты Sim комбинируются в одно значение). Лучшие результаты из 6 проведенных циклов обучения и валидации, полученные на валидационном наборе 3V, представлены в таблице 4.3.

Таблица 4.3. Значения показателей метрик для моделей

Метрика	Точная классификация (Четкая)			Нечеткая классификация		
	MAPE	nDCG @k	Accuracy	MAPE	nDCG @k	Accuracy
Модель						
М1	14%	61%	80%	более 50%	Менее 50%	70%
М2а	8.1%	81%	100%	13.4%	78.4%	100%
М2б	8.8%	81%	100%	11.5%	78.4%	100%
М3	13.4%	89%	90%	более 50%	Менее 50%	60%
Комбинирование результатов М2а и М3	10.6%	89%	100%	более 50%	Менее 50%	100%
Комбинирование результатов М2а и М2б	7.6%	81%	100%	9.4%	81%	100%
М4	9.5%	76%	100%	27.3%	Менее 50%	80%
Аддитивная свертка с постоянными коэффициентами важности элементов	11.7%	55.50%	95%	14%	60%	90%

На основе анализа данных таблицы можно сделать выводы, что для четкой и нечеткой классификации целесообразно применить комбинирование моделей, где итоговый результат $Sim(.)$ определяется в общем случае формулой вида: $Sim(.) = (1-\beta) Sim1 + \beta Sim2$, где $Sim1, Sim2$ – результат вычислений моделей 1 и 2 соответственно; $\beta \in [0,1]$ - подбираемый экспериментально коэффициент.

Для четкой классификации наилучшая модель с точки зрения верности ранжирования – ансамблированная модель M2a и M3. С точки зрения минимизации MAPE: ансамбль моделей M2a и M2b. В случае нечеткой классификации лучшее решение по показателям метрик MAPE и nDCG@k: ансамбль моделей M2a и M2b.

Оценка стабильности показателей в результате 6 проведенных циклов показала: модели M2a и M2b выдают стабильные показатели метрик на протяжении всех циклов (отклонение MAPE в пределах 5%, nDCG@k, Accuracy без отклонений), модель M3 показала значительное увеличение MAPE (до 25%) в 2-х из 6 циклов, модель M1 нестабильна: в 4 из 6 циклов показатели MAPE более 50%.

Данные результаты получены на сложных примерах, где разница значений Sim при ранжировании находится в пределах 10%. В таких случаях при сравнении с аддитивной сверткой нейросеть показывает лучшие показатели по метрикам, а также имитирует когнитивные функции человека, верно ранжируя ситуации по значению Sim .

На рисунке 4.8 представлен график значения Sim ситуаций из валидационного набора 3V.3. Сплошной линией представлен график значения Sim , полученный аддитивной сверткой для каждой сравниваемой ситуации из набора. Можно увидеть, что 4 первые ситуации имеют одинаковую схожесть со сравниваемой ситуацией, в то время как эксперт их проранжировал (значения графика точечной линии). Нейросеть смогла повторить выбор эксперта. На графике представлены значения Sim , полученные ансамблированной моделью M2a и M3 (пунктирная линия).

Ранжирование важно для дальнейшего извлечения программы действий из базы знаний. Чем более схожая ситуация будет выбрана, тем более подходящее решение будет извлечено.

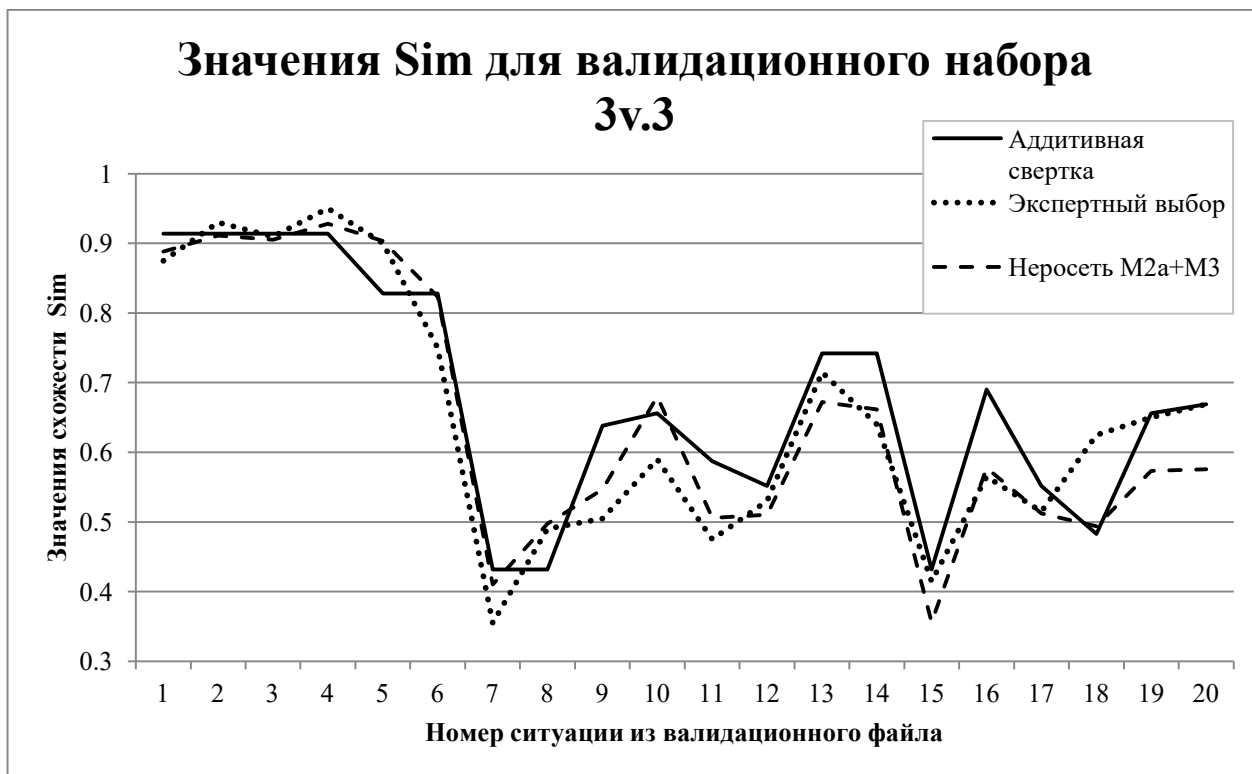


Рисунок 4.8 – График значения Sim сравниваемых ситуаций из набора 3V.3.

Стоит отметить, при оценке MAPE исключительно на парах ситуаций класса «похож» ($Sim > 0.8$) значения MAPE не превышают 5.5% у моделей M2 и их комбинаций.

В случае недостатка обучающих данных, например, в начале работы системы, будет уместно применение аддитивной свертки для определения схожести ситуаций. Не смотря на низкие показатели качества метрики ранжирования ($nDCG@k$), метрика отнесения к классу похож/непохож (Accuracy) имеет достаточно высокие значения, что показывает возможность с помощью простой аддитивной свертки отбирать похожие ситуации, хоть и без возможности их ранжирования.

Исследование набора с комбинированными метками.

низкого до высокого). Благодаря этому стало возможным сравнить для каждого входного S_{it} подмножество S_{it_j} , которое ранжируется по Y_v в сторону уменьшения при условии $Y_v \geq T$ (порог для оценки сходства).

В результате определения архитектуры (см. раздел 3.1) получена ансамблевая модель, архитектура которой изображена на рисунке 4.9.

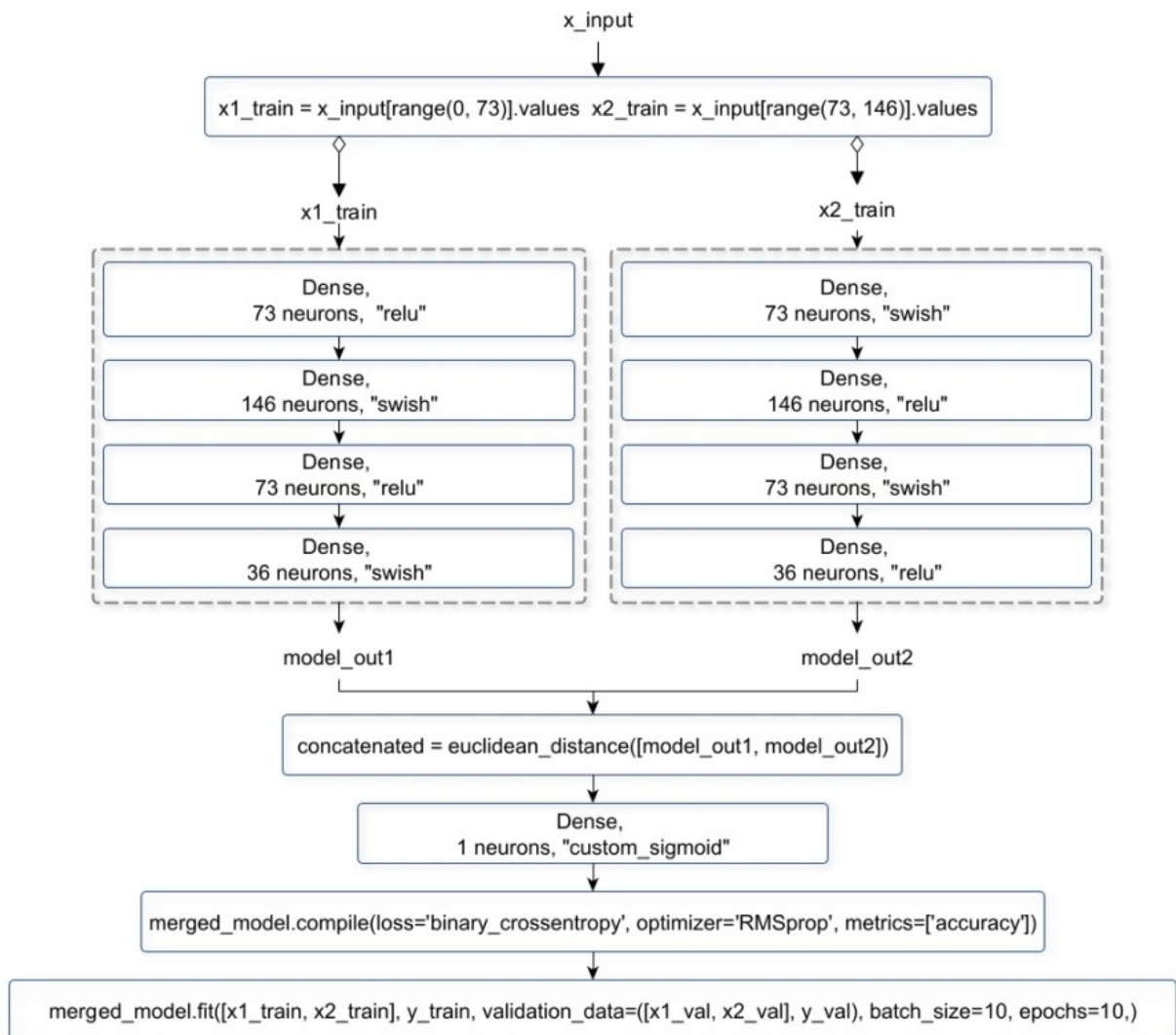


Рисунок 4.9 – Архитектура нейросетевой модели

Оценка эффективности обученной модели имеет следующий подход. Для каждого подмножества из набора валидационных данных были составлены пары последовательностей, ранжированных по степени сходства.

Первая последовательность содержит пары с вычисленным сходством, используя полученную модель нейронной сети. Вторая последовательность содержит пары со сходством, основанные на Y_v метках. Третья

последовательность содержит пары со сходством, рассчитанным с помощью модели взвешенной суммы, где сходство рассчитывается с помощью агрегирования локальных показателей сходства. Весовые коэффициенты локальных показателей сходства были установлены экспертом и не менялись.

При сравнении результатов использовались следующие метрики:

- метрика для оценки верности ранжирования k ситуаций лучших по величине Sim . Достигнуто значение 98% (на основе $nDCG@k$);

- метрика для оценки верности определения класса Accuracy. В рассматриваемом случае класс пар ситуаций «похож/непохож» при пороге $Sim > 0.5$ Достигнуто значение 100%.

Проведенные эксперименты показали, что в условиях малого объема обучающей выборки (150 пар ситуаций) модель показывает высокую точность при оценке схожести ситуаций на проверочных данных.

Стоит отметить, что при отборе ситуаций из базы недостаточно оценить класс «похож/непохож». Более важно оценивать способность модели делать правильный отбор и ранжирование лучших вариантов (ситуаций и решений), что соответствует практике человеко-машинного принятия решений.

В результате работы математической функции возникли коллизии, т.е. 4 ситуации были представлены одинаковыми значениями схожести, в то время как нейросетевая модель смогла их отранжировать, при этом ранги по схожести совпали с выбором эксперта.

Выводы по главе 4

Разработан комплекс программ, реализованный в виде Платформы для быстрой сборки интеллектуальных систем поддержки принятия решений, отличающийся микроядерной архитектурой, в частности, составом модулей и их взаимодействием, что позволяет собирать и настраивать из готовых модулей предметно-ориентированную ИСППР и осуществлять поддержку принятия решений в ходе эксплуатации на основе гибридного метода рассуждения на основе прецедентов.

Результаты главы опубликованы в работах [119-121]. Проведена апробация на примерах сложных технологических объектах: тепловой пункт здания (Представлено в разделе 4.4), электроподстанция [113] и умная ферма [120]. Результаты апробации показали применимость и эффективность предложенных методов для вывода состава решения в случае сложного технологического объекта, состояние которого описывается мультимодальными данными.

ЗАКЛЮЧЕНИЕ

В диссертации решена актуальная задача исследования и разработки математического, алгоритмического и программного обеспечения интеллектуальных систем поддержки принятия решений на сложных технологических объектах (СТО). Решение этой задачи имеет научную и практическую ценность для построения ИСППР при эксплуатации СТО, для которых значимы оценка комплексной ситуации на объекте с учетом контекста и быстрое предоставление пользователю предлагаемых к исполнению в случае нежелательной ситуации действий.

Предложена новая концепция моделирования сложного технологического объекта, где внешние условия (контекст) рассматриваются как часть этого объекта. На ее основе разработаны новые методы математического моделирования для представления сложного технологического объекта, ситуаций на объекте и решений, сопоставленных с проблемными, нежелательными ситуациями. Такой подход к моделированию позволяет единообразно представить различную информацию, поступающую от компонентов, для моделирования комплексной ситуации на объекте с учетом контекста.

Для выдачи решений предложен метод моделирования процесса выработки решения на базе гибридного метода рассуждения на основе прецедентов. Метод позволяет устранить ограничения раздельного применения этих подходов: системы, основанные на знаниях, требуют трудоемкой работы по выявлению и формализации знаний; методы машинного обучения требуют большие объемы обучающих примеров.

Для поиска схожих прецедентов и дальнейшей выдачи их решения введена обучаемая функция выбора решения на основе определения схожести ситуаций. Функция строится на основе нейросетевой модели, архитектура которой определяется в ходе подбора гиперпараметров на этапе обучения. Эта функция, в отличие от традиционных метрик близости, позволяет учитывать особенности функционирования конкретного СТО, так как соответствующие знания

используются при обучении модели, что определяет гибкость предложенного подхода к нахождению схожих ситуаций СТО.

Разработан алгоритм сборки и адаптации решения на основе имеющихся прецедентов в базе знаний. Этот алгоритм позволяет в случае отсутствия в базе знаний подходящего решения на основе имеющихся прецедентов синтезировать новое решение путем сборки решения из имеющихся в базе элементарных преобразований состояний компонентов объекта и последовательностей таких преобразований. Разработан алгоритм генерации обучающих множеств, опирающийся на предложенный метод моделирования состояний сложных технологических объектов. Совместная работа алгоритмов позволяет пополнять базу знаний новыми прецедентами на основе ранее валидированных знаний (решений и данных) об объекте.

На основе разработанных методов и алгоритмов разработан комплекс программ – Платформа для быстрой сборки интеллектуальных систем поддержки принятия решений, позволяющая собирать и настраивать из готовых модулей предметно-ориентированную ИСППР. Платформа представляет собой инструмент для создания ИСППР, предназначенных для конкретных СТО различных областей, что определяет универсальность предложенных решений.

В процессе исследования проведена апробация предложенных методов, алгоритмов и программного комплекса на примерах сложных технологических объектах: тепловой пункт здания, электроподстанция и умная ферма, где в качестве получаемых данных выступала, в том числе, текстовая информация от персонала в неструктурированном виде. Результаты апробации показали применимость и эффективность предложенных методов для вывода состава решения в случае сложного технологического объекта, состояние которого описывается мультимодальными данными. Результаты использованы при разработке системы поддержки принятия решений на объектах агрокомплекса, что подтверждается актом о внедрении (Приложение 4).

Ограничения разработанных методов и алгоритмов для интеллектуальных систем поддержки принятия решений сводятся к ограничениям ситуационного

метода (СВР): отсутствие ситуации (отсутствие информации для принятия решения). Другое ограничение заключается в невозможности формализовать информацию для получения описания комплексной ситуации, в том числе, вследствие ее неточности, абстрактности, субъективности. Также нецелесообразно применение в случае отсутствия достаточного объема знаний (пар ситуация-решение) и невозможности накопления этих знаний в ходе эксплуатации (расширения базы знаний).

Перспективы дальнейшей разработки темы. Перспективы связаны с внедрением языковых моделей с технологиями RAG (Retrieval Augmented Generation), визуально-лингвистических моделей, что позволит усовершенствовать этап идентификации актуального состояния сложного технологического объекта, когда в ходе диалога с пользователем на основе неточной субъективной информации уточняются значения атрибутов сложившейся ситуации и выдается соответствующее решение. Внедрение этого подхода потенциально значительно снизит трудозатраты на наполнение базы знаний, настройку функции выбора схожих ситуаций, а также снимет часть ограничений применимости.

СПИСОК ЛИТЕРАТУРЫ

1. Поспелов Д.А. Ситуационное управление. Теория и практика. - М.: Наука, 1986
2. Башлыков, А.А. Применение методов теории прецедентов в системах поддержки принятия решений при управлении трубопроводными системами // Автоматизация, телемеханизация и связь в нефтяной промышленности. – 2016. – № 1. – С. 23-33
3. Kuzyakov, O.N., Andreeva M.A. Applying Case-Based Reasoning Method for Decision Making in IIoT System. 2020 International Multi-Conference on Industrial Engineering and Modern Technologies. – 2020. – 9271301. DOI:10.1109/FarEastCon50210.2020.9271301
4. Юрин А.Ю. Методы группового выбора для адаптации решений, полученных в результате рассуждений на основе прецедентов // Искусственный интеллект и принятие решений. – 2013. – №3. – С.78-85.
5. Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений, 2009, №2.
6. Sayed K., Gabbar H. SCADA and smart energy grid control automation // Smart Energy Grid Engineering. – 2017. – С. 481–514. – DOI: 10.1016/B978-0-12-805343-0.00018-8.
7. Ковалев С. П. Применение цифровых двойников в автоматизированном управлении высокотехнологичным промышленным производством // Мехатроника, автоматизация, управление. – 2024. – Т. 25, № 4. – С. 211–220. – DOI: 10.17587/mau.25.211-220.
8. Jiang Z., Lv H., Li Y., Guo Y. A novel application architecture of digital twin in smart grid // Journal of Ambient Intelligence and Humanized Computing. – 2021. – DOI: 10.1007/s12652-021-03329-z.
9. Kiziroglou M. E., Boyle D. E., Wright S. W., Yeatman E. M. Acoustic power delivery to pipeline monitoring wireless sensors // Ultrasonics. – 2017. – Т. 77. – С. 54–60.

10. Zarifi M. H., Deif S., Daneshmand M. Wireless passive RFID sensor for pipeline integrity monitoring // *Sensors and Actuators A: Physical*. – 2017. – Т. 261. – С. 24–29.
11. Jia Z. Pipeline abnormal classification based on support vector machine using FBG hoop strain sensor // *Opti*. – 2018. – Т. 170. – С. 328–338.
12. Способ мониторинга предаварийного состояния контролируемых объектов / А. М. Винограденко [и др.]. - (Теория и принципы построения датчиков, приборов и систем). - Текст : непосредственный // *Датчики и системы*. - 2014. - № 9. - С. 8-14 : ил. - Библиогр.: с. 13-14 (17 назв.). - ISSN 1992-7185.
13. Zrelli A., Ezzedine T. Design of optical and wireless sensors for underground mining monitoring system // *Optik*. – 2018. – Т. 170. – С. 376–383.
14. Chen J., Zhou H., Hu H., Song Y., Huang Y. Research on agricultural monitoring system based on convolutional neural network // *Future Generation Computer Systems*. – 2018. – Т. 88. – С. 271–278.
15. He P., Wang J. Statistical process monitoring as a big data analytics tool for smart manufacturing // *Journal of Process Control*. – 2018. – Т. 67. – С. 35–43.
16. Некрасов, И. В. Предиктивная диагностика отказов энергогенерирующего оборудования на основе нейронной сети / И. В. Некрасов, Ю. Д. Константиновский, Н. С. Кукин // V Международная конференция по нейронным сетям и нейротехнологиям (NeuroNT'2024) : Сборник докладов конференции, Санкт-Петербург, 20 июня 2024 года. – Санкт-Петербург: Федеральное государственное автономное образовательное учреждение высшего образования Санкт-Петербургский государственный электротехнический университет ЛЭТИ им. В.И. Ульянова (Ленина), 2024. – С. 224-229.
17. Rukmani P., Teja G. K., Vinay S., Reddy B. P. K. Industrial Monitoring Using Image Processing, IoT and Analyzing the Sensor Values Using Big Data // *Procedia Computer Science*. – 2018. – Т. 133. – С. 991–997.
18. Сысоев Ю. С., Сальников А. А., Бекетов В. Г., Чернов А. В. Прогнозирование состояний технологических объектов на основе текущего мониторинга значений их параметров // *Измерительная техника*. – 2016. – № 4. – С. 3–7.

19. Chelaru E., Grigoras G. Decision Support System to Determine the Replacement Ranking of the Aged Transformers in Electric Distribution Networks // 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). – 2020. – DOI: 10.1109/ECAI50035.2020.9223122.
20. Dogdu E., Murat Ozbayoglu A., Benli O., Akinc H., Erol E., Atasoy T. et al. Ontology-centric data modelling and decision support in smart grid applications: a distribution service operator perspective // 2014 IEEE International Conference on Intelligent Energy and Power Systems (IEPS). – 2014. – DOI: 10.1109/IEPS.2014.6874179.
21. Использование современных информационных технологий для анализа доменного процесса / Н. А. Спирин, В. В. Лавров, В. Ю. Рыболовлев [и др.] // *Металлург*. – 2016. – № 5. – С. 13-19. – EDN WHJMVD.
22. Nina D., da Fonseca Neto J., Ferreira E., Miranda dos Santos A. Hybrid Support System for Decision Making Based on MLP-ANN, IED and SCADA for Disturbances Analysis of Electrical Power Distribution Transformers // 2013 UKSim 15th International Conference on Computer Modelling and Simulation. – 2013. – DOI: 10.1109/UKSim.2013.147.
23. Планирование системы ремонтов конвертеров в условиях квазипериодического функционирования агрегатов / М. Е. Корнет, А. В. Зимин, И. В. Буркова, В. В. Зимин // *Известия высших учебных заведений. Черная металлургия*. – 2024. – Т. 67, № 6. – С. 738–743. – DOI: 10.17073/0368-0797-2024-6-738-743. – EDN BQWVQA.
24. Шпицер В. Я., Кривин В. В., Толстов В. А. Предсказательное диагностирование на основе индикатора Херста и логистических трендов // *Глобальная ядерная безопасность*. – 2021. – № 4. – С. 47–55. – DOI: 10.26583/gns-2021-04-05.
25. Samoilova E. M. Building expert system of support of decision-making as intellectual component of the monitoring system of technological process // *Vestnik PNIPU. Mashinostroenie, materialovedenie*. – 2016. – Т. 2(18). – С. 128–141.

26. Chan W. C. An expert decision support system for monitoring and diagnosis of petroleum production and separation processes // *Expert Systems with Applications*. — 2005. — Т. 1(29). — С. 131–143.
27. Теория и методы создания интеллектуальных компьютерных систем / В. Л. Арлазаров [и др.] // *Информационные технологии и вычислительные системы*. — 1998. — № 1. — С. 3—13.
28. Осипов, Г. С. Динамика систем, основанных на знаниях / Г. С. Осипов // *Известия РАН. Теория и системы управления*. — 1998. — №5. — С. 24 —28.
29. Голицын, Г. А. Нейронные сети и экспертные системы: перспективы интеграции / Г. А. Голицын, И. Б. Фоминых // *Новости искусственного интеллекта*. — 1996. — № 4. — С. 121—146.
30. Turban, E. *Decision Support Systems and Intelligent Systems (7th Edition)* / E. Turban, J. E. Aronson, T.-P. Liang. — Prentice-Hall, Inc., 2004. 22
31. Haagsma, I. G. *Decision support systems: An integrated and distributed approach* / I. G. Haagsma, R. D. Johans // *Transactions on Ecology and the Environment*. — 1994. — С. 205—212.
32. Fox, J. *Safe and sound. Artificial intelligence in hazardous applications* / J. Fox, S. Das. — MIT Press, 2000. — 325 p. 12.
33. Концепция инженерии знаний в задачах обеспечения интероперабельности АСУ и информационных систем на основе интеллектуальных технологий / М. Ю. Охтилев, В. Н. Коромысличенко, П. А. Охтилев [и др.] // *Интеллектуальные технологии на транспорте*. — 2023. — № 3(35). — С. 5-13. — DOI 10.24412/2413-2527-2023-335-5-13. — EDN OFSKYM.
34. Hogan A., Blomqvist E., Cochez M., D’amato C., De Melo G., Gutierrez C., Kirrane S., Labra Gayo J. E., Navigli R., Neumaier S., Ngonga Ngomo A.-C., Polleres A., Rashid S. M., Rula A., Schmelzeisen L., Sequeda J., Staab S., Zimmermann A. *Knowledge Graphs* // *ACM Comput. Surv.* — 2021. — Т. 54, № 4. — Article 71. — С. 37. — DOI: 10.1145/3447772.
35. Решение технологических задач в доменном производстве с использованием модельных систем поддержки принятия решений / А. В. Павлов,

А. А. Полинов, Н. А. Спиринов, О.П. Онорин, Лавров В.В., Гурин И.А. // Сталь. – 2019. – № 3. – С. 13-20. – EDN KJLPEI.

36. Модельные системы поддержки принятия решений в АСУ ТП доменной плавки: монография / Н.А. Спиринов, В.В. Лавров, В.Ю. Рыболовлев, А.В. Краснобаев, О.П. Онорин, И.Е. Косаченко; под ред. Н.А. Спирина. – Екатеринбург: УрФУ, 2011. – 462 с. Электронный научный архив УрФУ (полная версия), URL: <http://hdl.handle.net/10995/39973> (Дата обращения: 20.02.2024).

37. Системы поддержки принятия решений для управления технологическими процессами в пирометаллургии: учебное пособие / Н.А. Спиринов, В.В. Лавров, И.А. Гурин; под общ. ред. Н.А. Спирина; Министерство науки и высшего образования РФ, Уральский федеральный университет. – 2-е изд., испр. и доп. – Екатеринбург: Агентство Маркетинговых Коммуникаций «День РА», 2024. – 308 с. – ISBN 978-5-6050813-2-6. (Рекомендовано методическим советом Уральского федерального университета для студентов вуза, обучающихся по направлениям подготовки 09.04.02 – Информационные системы и технологии, 22.04.02 – Metallургия, протокол заседания Методического совета УрФУ №20 от 10.12.2024 г.). Электронный научный архив УрФУ, URL: <https://elar.urfu.ru/handle/10995/140113> (Дата обращения: 20.02.2024)

38. Комплексное моделирование, автоматизация и интеллектуализация проактивного управления жизненным циклом сложных объектов / Б. В. Соколов, Р. М. Юсупов, М. Ю. Охтилев, П. А. Охтилев // Имитационное моделирование. Теория и практика (ИММОД-2023) : Сборник трудов одиннадцатой всероссийской научно-практической конференции по имитационному моделированию и его применению в науке и промышленности, Казань, 18–20 октября 2023 года. – Казань: Издательство АН РТ, 2023. – С. 65-76. – EDN OVCSBH.

39. Вивчарь, Р. М. Методика многокритериального оценивания эффективности функционирования стохастических сложных технических систем / Р. М. Вивчарь, А. И. Птушкин, Б. В. Соколов // Авиакосмическое приборостроение. – 2022. – № 7. – С. 3-14. – DOI 10.25791/aviakosmos.7.2022.1286. – EDN XCYDAI.

40. Pospelov D. A. Situation Control: an Overview// Proceedings of Workshop on Russian Situation Control and Cybernetic/Semiotic Modeling. Ed. by R. J. Strohn. Columbus. Battelle, 1996, pp. 7–37.
41. Improving case-based reasoning with the aid of multi-criteria and group decision-making methods/ Galina S. Maltugueva, Alexander Yu. Yurin // DOI: 10.23919/MIPRO.2019.8756874, 2019.
42. Hybrid Decision Support System for automating decision making in the event of defects in the era of Zero Defect Manufacturing/Foivos Psarommatis , Dimitris Kiritsis // Journal of Industrial Information Integration, 2021
43. DecisionMaster: A multi-criteria decision support system with ability to combine different decision rules / Aleksandr Podvesovskii, Alena Zakharova, Dmitriy Korostelyov, Aleksandr Kuzin // SoftwareX 16 (2021) 100814.
44. A decision support system for urban infrastructure inter-asset management employing domain ontologies and qualitative uncertainty-based reasoning/ Lijun Wei, Heshan Du et al.//Expert Systems with Applications, 2020.
45. Development of an intelligent decision support system to control the process of well drilling under complicated conditions/V. Shavranskyi, G. Sementsov// Information And Controlling System, 2019, DOI: 10.15587/1729-4061.2019.179401
46. Антонов, В. В. Усовершенствование ситуационной методологии разработки систем поддержки принятия решений для предприятий / В. В. Антонов, К. А. Конев // Онтология проектирования. – 2022. – Т. 12, № 4(46). – С. 547-561. – DOI 10.18287/2223-9537-2022-12-4-547-561.
47. Пальчевский, Е. В. Методы нейросетевой обработки больших темпоральных данных для информационной поддержки принятия управленческих решений (на примере электроэнергетики): диссертация на соискание ученой степени кандидата технических наук / Пальчевский Евгений Владимирович, 2024. – 194 с.
48. Gary M. Deep Learning: A Critical Appraisal // Preprint. New York University, 2017.

49. Ситуационное управление энергетическими объектами и процессами электроэнергетической системы: [монография] / Ю. А. Секретарев и др.] ; И. Л. Кескевич. – Новосибирск : НГТУ, 2007. – 306 с. – (Монографии НГТУ). – ISBN 978-5-7782-0746-2.
50. Виттих, В. А. Проблемы управления и моделирования в сложных искусственных системах / В. А. Виттих // Мехатроника, автоматизация, управление. – 2010. – № 12. – С. 17-23.
51. Филимонов, А. Б. Ситуационный подход в задачах автоматизации управления техническими объектами / А. Б. Филимонов, Н. Б. Филимонов // Мехатроника, автоматизация, управление. – 2018. – Т. 19, № 9. – С. 563-578. – DOI 10.17587/mau.19.563-578. – EDN XZPFAD.
52. Case Based Reasoning as a Model for Cognitive Artificial Intelligence/Susan Craw, Agnar Aamodt // ICCBR 2018.
53. Eremeev A., Varshavskiy P., Alekhin R. Case-Based Reasoning Module for Intelligent Decision Support Systems // Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’16). – 2016. – Т. 1. – Ч. III. – С. 207–216. – DOI: 10.1007/978-3-319-33609-1_18.
54. Eremeev A. P., Varshavskij P. R., Kurilenko I. E. Modeling time dependencies in intelligent decision support systems on precedents // Information Technologies & Knowledge, 3(6), 227-239 (2012).
55. Ramon López de Mántaras et al. Retrieval, reuse, revision, and retention in case-based reasoning // The Knowledge Engineering Review, 2005, Vol. 00:0, 1–2.
56. Huang K., Nie W., Luo N. Scenario-based marine oil spill emergency response using hybrid deep reinforcement learning and case-based reasoning // Applied Science. – 2020. – Т. 10, № 15. – С. 5269. – DOI: 10.3390/app10155269.
57. Jiang X., Wang S., Wang J., Lyu S., Skitmore M. A Decision Method for Construction Safety Risk Management Based on Ontology and Improved CBR: Example of a Subway Project // International Journal of Environmental Research and Public Health. – 2020. – Т. 17. – С. 3928. – DOI: 10.3390/ijerph17113928.

58. Kouissi M., En-naimi El M., Zouhair A. Hybrid solution for wind turbines power curve modeling founded on case-based reasoning, multi-agent system and the k-nearest neighbors algorithm // *Journal of Theoretical and Applied Information Technology*. – 2021. – Т. 99. – № 12.
59. Inference and reasoning in a Bayesian knowledge-intensive CBR system / Hoda Nikpour, Agnar Aamodt // *Progress in Artificial Intelligence* <https://doi.org/10.1007/s13748-020-00223-1>
60. Twin-Systems to Explain Artificial Neural Networks using Case-Based Reasoning: Comparative Tests of Feature-Weighting Methods in ANN-CBR Twins for XAI / Eoin M. Kenny, Mark T. Keane // *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. – 2019.
61. Mathisen B. M., Bach K., Aamodt A. Using extended siamese networks to provide decision support in aquaculture operations // *Applied Intelligence*. – 2021. – DOI: 10.1007/s10489-021-02251-3.
62. Hoffmann M., Bergmann R. Informed Machine Learning for Improved Similarity Assessment in Process-Oriented Case-Based Reasoning // *arXiv:2106.15931*. – 2021.
63. Leake D., Ye X., Crandall D. Supporting Case-Based Reasoning with Neural Networks: An Illustration for Case Adaptation // *CEUR-WS.org*. – 2021. – Т. 2846.
64. Zhai Z., Martinez J. F., Lucas Martinez N., Hernandez Diaz V. Applying case-based reasoning and a learning-based adaptation strategy to irrigation scheduling in grape farming // *Computers and Electronics in Agriculture*. – 2020. – Т. 178. – DOI: 10.1016/j.compag.2020.105731.
65. Борисов, В.В. Мониторинг и адаптация базы проектных прецедентов при управлении инновационными проектами на основе нечёткого онтологического подхода / В.В. Борисов, М.В. Черновалова, С.П. Курилин // *Онтология проектирования*. – 2020. – Т.10, №4(38). - С.516–526. – DOI: 10.18287/2223-9537-2020-10-4-516-526.
66. Solving the motion planning problem using learning experience through case-based reasoning and machine learning algorithms/ Mustafa F. Abdelwahed, Amr E. Mohamed, Mohamed Aly Saleh // *Ain Shams Engineering Journal* 11, 2020, 133–142.

67. Machine learning based feature selection and knowledge reasoning for CBR system under big data/ Yuan Guo, Bing Zhang, Y. Sun, K. Jiang, K. Wu // *Pattern Recognition* 112, 2021, 107805.
68. How Case-Based Reasoning Explains Neural Networks: A Theoretical Analysis of XAI Using Post-Hoc Explanationby-Example from a Survey of ANN-CBR Twin-Systems/Mark T. Keane, Eoin M. Kenny// *arXiv:1905.07186* , 2019.
69. Грибова В. В., Ковалев Р. И., Окунь Д. Б. Система назначения персонифицированного лечения по аналогии на основе гибридного способа извлечения прецедентов // *Программные продукты и системы*. – 2023. – Т. 36. – № 3. – С. 486–492. – DOI: 10.15827/0236-235X.142.486-492.
70. Грибова В. В., Гельцер Б. И., Шахгельдян К. И., Петряева М. В., Шалфеева Е. А., Костерин В. В. Гибридная технология оценки рисков и прогнозирования в кардиологии // *Врач и информационные технологии*. – 2022. – № 3. – С. 24–35. – DOI: 10.25881/18110193_2022_3_24.
71. Creation of Prototypes of Case-Based Knowledge Bases Using Transformations of Decision Tables to Predict the Risk of Forest Fires / N. O. Dorodnykh, O. A. Nikolaychuk, J. V. Pestova, A. Yu. Yurin // *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications*. – 2023. – Vol. 33, No. 3. – P. 274–281. – DOI 10.1134/S1054661823030094. – EDN SZADVG.
72. Карпов Л. Е., Юдин В. Н. Адаптивное управление по прецедентам, основанное на классификации состояний управляемых объектов // *Труды Института системного программирования РАН*. – 2017. – Т. 13, № 2. – С. 37.
73. Louati A., Louati H., Li Z. Deep learning and case-based reasoning for predictive and adaptive traffic emergency management // *The Journal of Supercomputing*. – 2021. – Т. 77. – С. 4389–4418. – DOI: 10.1007/s11227-020-03435-3.
74. Perner P. Case-Based Reasoning – Methods, Techniques, and Applications // *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Application. CIARP 2019: Havana, Cuba*. – 2019. – С. 16–30. – DOI: 10.1007/978-3-030-33904-3_2.

75. Gabel T., Godehardt E. Top-Down Induction of Similarity Measures Using Similarity Clouds // Proceedings of the International Conference on Case-Based Reasoning. – 2015. – С. 149–164. – DOI: 10.1007/978-3-319-24586-7_11.
76. Mathisen M. B., Bach K., Aamodt A. Using extended siamese networks to provide decision support in aquaculture operations // Applied Intelligence. – 2021. – DOI: 10.1007/s10489-021-02251-3.
77. Nikpour H., Aamodt A. Inference and reasoning in a Bayesian knowledge-intensive CBR system // Progress in Artificial Intelligence. – 2021. – Т. 10. – С. 49–63. – DOI: 10.1007/s13748-020-00223-1.
78. Aha D. W. Case-based learning algorithms // Proceedings of the 1991 DARPA Case-Based Reasoning Workshop. – Washington, USA. – 1991. – Т. 1. – С. 147–158.
79. Chen D., Burrell P. Case-based reasoning system networks: A review // Neural Computing Applications. – 2001. – Т. 10. – С. 264–276.
80. Kenny E. M., Keane M. T. Twin-Systems to Explain Artificial Neural Networks using Case-Based Reasoning: Comparative Tests of Feature-Weighting Methods in ANN-CBR Twins for XAI // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19). – 2019. – С. 2708–2715. – DOI: 10.24963/ijcai.2019/376.
81. Mathisen M. B., Aamodt A., Bach K., Langseth H. Learning similarity measures from data // Progress in Artificial Intelligence. – 2020. – Т. 9. – С. 129–143. – DOI: 10.1007/s13748-019-00201-2.
82. Караневская, Т. Н. Моделирование технологических процессов в целях алгоритмизации задачи управления объектами промышленной подготовки нефти / Т. Н. Караневская, А. Г. Шумихин // Известия высших учебных заведений. Серия: Химия и химическая технология. – 2020. – Т. 63, № 2. – С. 84-90. – DOI 10.6060/ivkkt.20206302.6100. – EDN YSJMCE.
83. Масляницын, А. П. Математическое моделирование радиатора системы отопления как объекта управления / А. П. Масляницын, Е. В. Масляницына, М. С. Краснова // Градостроительство и архитектура. – 2021. – Т. 11, № 3(44). – С. 38-43. – DOI 10.17673/Vestnik.2021.03.06. – EDN KHBAIT.

84. Охотников, А. Л. Ситуационное семиотическое управление / А. Л. Охотников, А. А. Павловский // Наука и технологии железных дорог. – 2020. – Т. 4, № 3(15). – С. 53-62. – EDN VBJOFJ.
85. Антипин, А. Ф. Моделирование технологических процессов и объектов на основе нечёткой логики с использованием многомерных интервально-логических регуляторов / А. Ф. Антипин, Е. В. Антипина // Автометрия. – 2020. – Т. 56, № 1. – С. 83-92. – DOI 10.15372/AUT20200109. – EDN NJGIUC.
86. Craw S., Wiratunga N., Rowe R. C. Learning adaptation knowledge to improve case-based reasoning // Artificial Intelligence. – 2006. – Т. 170. – С. 1175–1192. – DOI: 10.1016/j.artint.2006.09.001.
87. Yurin A. Yu. Application of group decision-making methods for case adaptation // Scientific and Technical Information Processing. – 2013. – № 3. – С. 78–85.
88. Chebel-Morello B., Haouchine M. K., Zerhouni N. Reutilization of diagnostic cases by adaptation of knowledge models // Engineering Applications of Artificial Intelligence. – 2013. – Т. 26. – С. 2559–2573.
89. Zhai Z., Martínez J. F., Martínez N. L., Díaz V. H. Applying case-based reasoning and a learning-based adaptation strategy to irrigation scheduling in grape farming // Computers and Electronics in Agriculture. – 2020. – Т. 178. – № 105741. – DOI: 10.1016/j.compag.2020.105741.
90. Michel T. Machine Learning. – New York: McGraw Hill, 1997.
91. Bishop C. M. Pattern Recognition and Machine Learning. – Springer, 2006.
92. Глухих И. Н., Карякин Ю. Е., Березовский А. К., Дудина А. А. Классификация состояний объектов городской инфраструктуры с использованием нейронных сетей // Инженерный вестник Дона. – 2021. – № 8.
93. Akhmadulin R. K., Gluhih I. N., Karyakin I. Y. An object-oriented model of case-based reasoning system using situations tree // 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT). – Baku, Azerbaijan. – 2016. – С. 1–4. – DOI: 10.1109/ICAICT.2016.7991668.
94. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016. – 326 с.

95. Accelerate Machine Learning with the cuDNN Deep Neural Network Library // Nvidia. Developer. URL: <https://developer.nvidia.com/blog/accelerate-machine-learning-cudnn-deep-neural-network-library/> (дата обращения: 14.05.2022).
96. Козлов, В. А. Электроснабжение городов / В. А. Козлов. – Л. : Энергоатомиздат. - 1988. – 293 с.
97. Водоснабжение. Проектирование систем и сооружений. Т. 2. Очистка и кондиционирование природных вод / под общ. ред. М. Г. Журба. - Вологод. гос. ун-т; НИИ ВОДГЕО. - М.; Вологда, 2001. - 324 с.
98. Газоснабжение / Брюханов О.Н., Жила В.А., Плужников А.И. М.: Академия, 2008, — 448 с., ISBN 978-5-7695-2595-7, ил.
99. Современные системы отопления / В. И. Назарова. — М. : РИПОЛ классик, 2011. — 320 с. : ил. — (Энциклопедия строительства).
100. Костюков, А. Ф. Анализ состояния электропроводок на основе теплотехнической модели / А. Ф. Костюков // Вестник Алтайского государственного аграрного университета. – 2015. – № 2(124). – С. 106-112. – EDN THKSIH.
101. Урумбаева, О. Б. Концепция интеллектуального управления энергосетью / О. Б. Урумбаева, Т. А. Шалаев, О. М. Шикунская // Инженерно-строительный вестник Прикаспия. – 2020. – № 3(33). – С. 69-74. – DOI 10.35108/isvp20203(33)69-74. – EDN VTKAZG.
102. Магарас, Ю. И. Прогнозирование погодных и дорожных условий как элемент интеллектуальной транспортной системы / Ю. И. Магарас, А. А. Клясова, А. В. Добринский // Вестник НЦБЖД. – 2017. – № 1(31). – С. 28-37. – EDN ZFTIVB.
103. Martin M., Molin E. Environmental Assessment of an Urban Vertical Hydroponic Farming System in Sweden // Sustainability. – 2019. – Т. 11. – № 4124. – DOI: 10.3390/su11154124.
104. Glukhikh D. Case Based Reasoning for managing urban infrastructure complex technological objects // CEUR Workshop Proceedings. – 2021. – Т. 2843. – № 038.

105. Glukhikh I., Chernysheva T., Glukhikh D. Neural Network Models for Situation Similarity Assessment in hybrid-CBR // *Journal of Intelligent & Fuzzy Systems.* – 2023. – С. 1–14. – DOI: 10.3233/JIFS-221335.
106. Glukhikh, I., Glukhikh, D. Case-Based Reasoning with an Artificial Neural Network for Decision Support in Situations at Complex Technological Objects of Urban Infrastructure // *Applied System Innovation.* — 2021. — Т. 4, № 4. — С. 73. DOI: 10.3390/asi4040073.
107. Добрынин, А. С. Прецедентный подход к управлению инцидентами в автоматизированных системах управления технологическими процессами // *Программные системы и вычислительные методы.* 2020. № 2. С. 45-52. DOI: 10.7256/2454-0714.2020.2.31040.
108. Еремеев А.П., Варшавский П.Р., Поляков С.А. Программная реализация модуля анализа данных на основе прецедентов для распределенных интеллектуальных систем // *Программные продукты и системы.* 2021. Т. 34. № 3. С. 381–389. DOI: 10.15827/0236-235X.135.381-389.
109. Chen M., Qu R., Fang W. Case-based reasoning system for fault diagnosis of aero-engines // *Expert Systems with Applications.* 2022. Vol. 202. 117350, DOI: 10.1016/j.eswa.2022.117350.
110. Асяев, Г. Д. Модели предиктивной защиты информации автоматизированной системы управления водоснабжением на основе временных рядов с использованием технологий машинного обучения // *Вестник УрФО. Безопасность в информационной сфере.* 2021. № 4(42). С. 39-45. DOI: 10.14529/secur210404.
111. Подопрсветов А.В., Анохин Д.А., Кий К.И., Орлов И.А. Сравнительный анализ метода геометризованных гистограмм и нейросетевого метода для распознавания дорожной разметки // *Препринты ИПМ им. М.В. Келдыша.* 2021. № 104. С. 1-22. DOI: 10.20948/prepr-2021-104.
112. Глухих, Д. И. Исследование нейросетевой модели для оценки схожести ситуаций в системе поддержки принятия решений на основе метода CBR / Д. И. Глухих // *Новые информационные технологии в нефтегазовой отрасли и*

образовании: Материалы XI Международной научно-практической конференции-конкурса, Тюмень, 18–19 апреля 2024 года. – Тюмень: Тюменский индустриальный университет, 2024. – С. 42-46. – EDN HPHQGR.

113. Glukhikh D., Shchinnikov I., Glukhikh I. Using hybrid-CBR for intelligence monitoring and decision-making systems on SMART grid // *Intelligent Decision Technologies* 2022, vol. 16, no. 2, pp. 449-456. DOI:10.3233/IDT-210239

114. Глухих, И. Н., Глухих, Д. И., Карякин, Ю. Е. Представление и отбор ситуаций на сложном технологическом объекте в условиях неопределенности // *Вестник Российского нового университета. Сер. Сложные системы: модели, анализ и управление.* — 2021. — № 2. — С. 65–73. DOI: 10.25586/RNU.V9187.21.02.P.065.

115. Глухих, И. Н., Глухих, Д. И., Карякин, Ю. Е. Нейросетевая архитектура вывода решений в опасных ситуациях на сложном технологическом объекте // *Прикладная информатика.* — 2021. — № 5. DOI: 10.37791/2687-0649-2021-16-5-99-107.

116. Глухих, И. Н., Глухих, Д. И. Адаптация решений системы с выводом на прецедентах при возникновении новых ситуаций [Текст] / И. Н. Глухих, Д. И. Глухих // *Южно-сибирский научный вестник.* — 2022. — № 2 (42). — С. 132–136.

117. Chen H., Birkelund Y., Zhang Q. Data-augmented sequential deep learning for wind power forecasting // *Energy Conversion and Management.* – 2021. – Т. 248. – № 114790. – DOI: 10.1016/j.enconman.2021.114790.

118. Wright S. J. Coordinate descent algorithms // *Mathematical Programming.* – 2015. – Т. 151, № 1. – DOI: 10.1007/s10107-015-0892-3.

119. Глухих, И. Н., Глухих, Д. И. Алгоритмы генерации обучающих множеств в системе с прецедентным выводом на основе ситуаций-примеров [Текст] / И. Н. Глухих, Д. И. Глухих // *Программные продукты и системы (Software&Systems).* — 2022. — № 4. — С. 660–669. DOI: 10.15827/0236-235X.140.660-669.

120. Глухих, И. Н. Сравнение и отбор ситуаций в системах вывода решений на прецедентах для «умной» фермы / И. Н. Глухих, А. С. Прохошин, Д. И. Глухих //

Информатика и автоматизация. – 2023. – Т. 22, № 4. – С. 853-879. – DOI 10.15622/ia.22.4.6. – EDN LEARPX.

121. Глухих И.Н., Глухих Д.И. Архитектура системы поддержки принятия решений на сложном технологическом объекте на основе гибридного CBR // В сборнике: Новые информационные технологии в нефтегазовой отрасли и образовании. Материалы IX Международной научно-практической конференции-конкурса. Отв. редакторы О.Н. Кузяков, О.В. Баяк. Тюмень, 2022. С. 99-102.

Приложение 1

**Платформа для быстрого создания интеллектуальных систем поддержки
принятия решений "ГИББС СИСТЕМ"**

Руководство пользователя

Листов 25

АННОТАЦИЯ

Программа предназначена для быстрого создания прикладных интеллектуальных систем поддержки принятия решений (ИСППР) путем их быстрой сборки из готовых модулей с настройкой на предметную область – дообучением необходимых моделей и баз знаний.

Программа может использоваться в процессах эксплуатации и управления сложными технологическими объектами. Тип ЭВМ: РС. ОС: Windows 10 и выше.

Функциональные возможности:

- обеспечение создания предметно-ориентированной интеллектуальной системы вывода решений на прецедентах в ситуациях, возникающих на сложных технологических объектах
- сборка функциональных модулей в готовую систему вывода решений;
- заполнение и ведение базы правил логического вывода;
- ведение базы прецедентов в виде пар «Ситуация-Решение»;
- классификация состояний элементов сложного объекта по входным данным;
- вывод решения по входным данным о ситуации.

СОДЕРЖАНИЕ

1. Общие сведения	125
2. Регистрация элементов	126
3. Обучение классификаторов	127
4. Обучение отбора ситуации из базы.....	138
5. Описание модулей классификации состояний	140
6. Ситуационная база знаний	147

1. ОБЩИЕ СВЕДЕНИЯ

В данном руководстве пользователь – это лицо, которому предоставлены возможности редактировать ситуационную базу знаний, базу правил, настраивать ИСППР (регистрировать элементы, настраивать классификаторы и модели оценки схожести).

1.1. Обозначение и наименование программы

Платформа для быстрого создания интеллектуальных систем поддержки принятия решений "ГИББС СИСТЕМ".

1.2. Программное обеспечение, необходимое для функционирования программы

Windows 10 и выше, интернет-браузер, Python 3.9

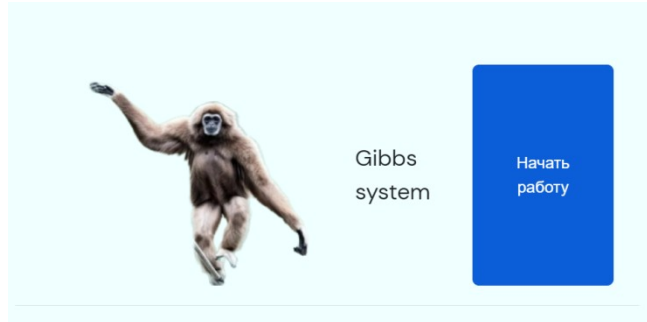
1.3. Языки программирования, на которых написана программа

Python, java

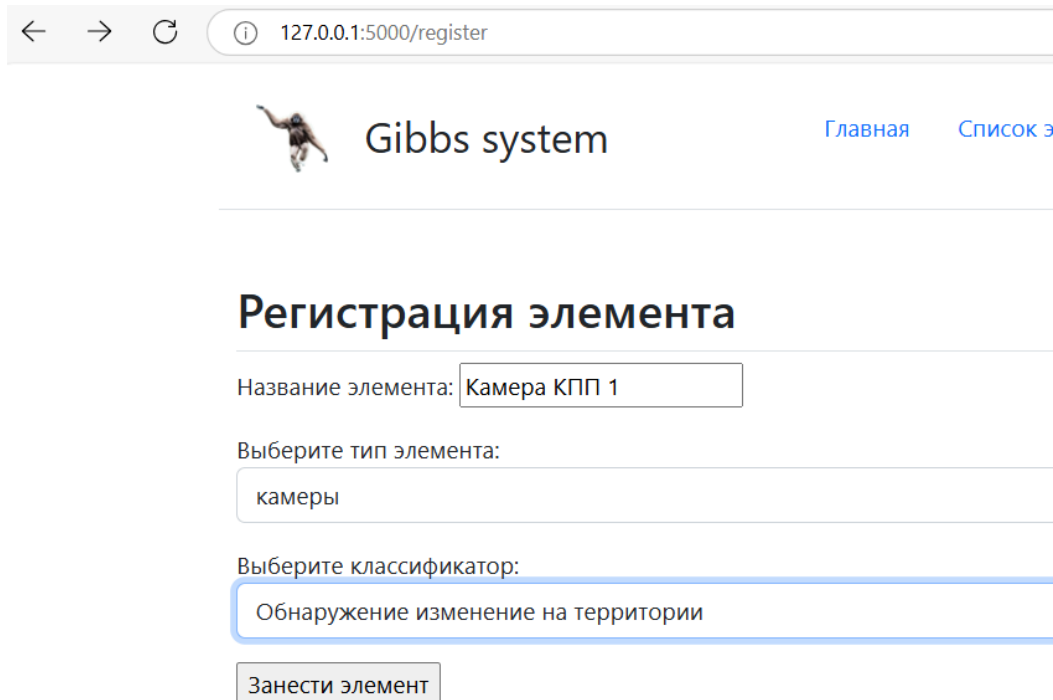
Языки разметки: html, css.

2. РЕГИСТРАЦИЯ ЭЛЕМЕНТОВ


1. Запустить на сервере приложение
2. Открыть браузер и перейти на страницу приложения по адресу сервера
3. Нажать кнопку «Начать работу» на главной странице



4. Перейти на вкладку «Регистрация элемента»



← → ↻ ⓘ 127.0.0.1:5000/register

 Gibbs system [Главная](#) [Список э](#)

Регистрация элемента

Название элемента:

Выберите тип элемента:

Выберите классификатор:

5. Воспользовавшись формой, зарегистрировать элемент. В окне «Название элемента» необходимо ввести название элемента. Необходимо выбрать тип элемента из списка для визуальной сортировки элементов: оборудование, персонал, камеры. Необходимо выбрать классификатор, исходя из типа данных, обеспечивающих классификацию.

3. ОБУЧЕНИЕ КЛАССИФИКАТОРОВ

Для обучения классификаторов на вкладке «Список элементов» выбрать элемент и перейти по ссылке «Настройка классификатора».

nasosN1	nasosN2	nasosN3
Актуальное состояние	Актуальное состояние	Актуальное состоян
сломано	работает	Набор мощности
Настройка классификатора	Настройка классификатора	Настройка классиф
Удалить элемент	Удалить элемент	Удалить элемент

При переходе по ссылке «Настройка классификатора» в каталоге education каталога GIBBS создается каталог с названием, соответствующим id элемента, где будут храниться файлы, необходимые для работы классификатора. При повторном переходе по ссылке «Настройка классификатора» каталог не перезаписывается и не создается повторно.

3.1. Обучение классификатора на экспертных правилах

При переходе по ссылке «Настройка классификатора» для элемента с классификатором на экспертных правилах откроется страница обучения классификатора.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/rules/1'. The page title is 'Gibbs system'. The navigation menu includes 'Главная', 'Список элементов', 'Регистрация элемента', and 'Состояя'. The main content area contains the following text and form fields:

Внесите правила классификации определенного состояния для элемента: pasosN1
 Внесенные состояния: Работает; Плохо работает; Сломано; Набор мощности

Состояние:

Выберите классификатор:

Параметры через запятую:

Значения параметров в диапазонах через запятую:

The bottom navigation menu includes 'Главная', 'Список элементов', 'Регистрация элемента', 'Состояния элементов', and 'Настр'. The footer text is '© ООО СТЭЙТИНТЕЛ, 2022'.

Здесь отображается имя настраиваемого элемента и уже внесенные состояния.

Состояния вносятся по одному путем внесения наименования, параметров и дальнейшего нажатия кнопки «Занести состояние», которая заносит настроенное состояние в базу.

1) В первой строке ввода необходимо ввести наименование одного состояния, для которого будут настроены параметры.

2) Далее возможно дополнительно выбрать условие выполнения:

- строгое выполнение: для идентификации состояния все фактические значения параметров должны совпасть с указанными (Например, для состояния «Исправно» указаны параметры t,p – температура, давление. Указаны диапазоны для них: 10-60, 150-300 соответственно. Если оба параметра находятся в указанных диапазонах, то идентифицируется состояние «Исправно»);

- нестрогое выполнение: хотя бы один параметр из списка находится в указанном диапазоне (Например, для состояния «Опасность» указаны параметры t,p

– температура, давление. Указаны диапазоны для них: 60-80, 300-400 соответственно. Если хотя бы один параметр находится в указанных диапазонах, то идентифицируется состояние «Опасность»).

3) Необходимо внести имена параметров через запятую в соответствии с их именами из файла источника данных. Например, t,p,a.

4) Необходимо внести диапазон значений параметров через запятую в порядке соответственно их именам. Например, 10-20,60-80,1300-1600 (соответственно, t: 10-20, p: 60-80, a: 1300-1600).

Вносить можно диапазоны чисел вида: «А-В» (более А не включительно, менее В включительно) или «>С» (более С не включительно) или «<D» (менее D включительно).

Вносить можно текстовые и числовые логические значения вида «=D» (проверка, равно ли актуальное значение D). Допускаются числа, текста (пример: =on, =150).

5) После заполнения полей необходимо нажать кнопку «Занести состояние». Параметры и их значения сохраняются в базе.

6) После занесения всех правил необходимо уйти с вкладки. Дополнить правила можно в любой момент.

Пример состава источника данных в формате .csv для нескольких элементов, где первая строка содержит имена столбцов с именами параметров:

name,t,p,a
nasosN1,25,85,off
nasosN2,16,75,on
nasosN3,8,50,on
nasosN4,8,50,on

3.2. Обучение классификатора на экспертных правилах «Проверка логических значений»

При переходе по ссылке «Настройка классификатора» для элемента с классификатором на экспертных правилах «Проверка логических значений» откроется страница обучения классификатора.

Здесь отображается имя настраиваемого элемента и уже внесенные состояния.

Состояния вносятся по одному путем внесения наименования, параметров и дальнейшего нажатия кнопки «Занести состояние», которая заносит настроенное состояние в базу.

Классификатор предполагает учет только двух возможных состояний, описывающихся в файле .json значениями 0/1. Настройка классификатора предполагает внесение только названий состояний.

Состав файла .json – источник данных для классификатора «Проверка логических значений» (пример):

```

{"state": 1
}

```

Внести имена состояний необходимо через запятую.

После внесения имен необходимо уйти с вкладки. Заменить имена можно в любой момент.

3.3. Обучение классификатора на основе нейросети «Нейросеть - табличные данные нормализованные»

При переходе по ссылке «Настроить классификатор» для элемента с классификатором на экспертных правилах «Нейросеть - табличные данные нормализованные» откроется страница обучения классификатора.

1) Необходимо загрузить набор данных для обучения, набор данных для проверки (опционально).

Набор данных должен содержать: нормализованные параметры через запятую, значение класса состояния. Пример состава обучающего набора представлен ниже:

1,1,0,1,0,1,1
1,2,1,1,1,1,0 ,

где первые 6 символов, означают некоторые параметры, последний символ – класс состояния.

2) Нажать кнопку загрузить. Начнется обучение нейросети. По окончании обучения откроется вкладка. Далее произойдет перенаправление на следующий вкладку для внесения имен состояний.

3) В открывшемся окне необходимо внести имена состояний.

The screenshot shows a web browser window with the address bar containing '127.0.0.1:5000/nnwtablestates/5'. The page header features the 'Gibbs system' logo and navigation links: 'Главная', 'Список элементов', 'Регистрация элемента', and 'Состояния'. The main content area is titled 'Регистрация состояний' and displays the message 'Обнаружено 5 состояний'. Below this, it instructs the user: 'Введите имена состояний через запятую согласно обучающему набору данных'. A text input field labeled 'Названия состояний:' contains the text '1,2,3,4,5'. A button labeled 'Занести элемент' is positioned below the input field. At the bottom of the page, there are additional navigation links: 'Главная', 'Список элементов', 'Регистрация элемента', 'Состояния элементов', and 'Настройки', along with a copyright notice: '© ООО СТЭЙТИНТЕЛ, 2022'.

После обучения система укажет, сколько обнаружено классов на основе обучающего набора данных, и предложит внести имена для всех классов через запятую.

В случае пропуска шага, имена состояний примут числовой порядок $1,2,3\dots n$, где n – количество обнаруженных классов.

3.4. Обучение классификатора на основе нейросети «Нейросеть - табличные данные ненормализованные»

При переходе по ссылке «Настроить классификатор» для элемента с классификатором на экспертных правилах «Нейросеть - табличные данные ненормализованные» откроется страница обучения классификатора.

Обучающий набор данных Не выбран ... один файл

Набор данных для проверки Не выбран ... один файл

Главная Список элементов Регистрация элемента Состояния элементов Настро

© ООО СТРЕЙТИНТЕЛ, 2022

1) Необходимо загрузить набор данных для обучения, набор данных для проверки (опционально).

Набор данных должен содержать: нормализованные параметры через запятую, значение класса состояния. Пример состава обучающего набора представлен ниже:

```
290,1.4,4,83,1.33,202,1,0
261,1.1,4,49,0.01,229,1,0
186,3.7,7,79,0.49,235,1,1 ,
```

где первые 7 значений, означают некоторые параметры, последний символ – класс состояния.

2) Нажать кнопку загрузить. Начнется обучение нейросети. По окончании обучения откроется вкладка. Далее произойдет перенаправление на следующий вкладку для внесения имен состояний.

3) В открывшемся окне необходимо внести имена состояний.

The screenshot shows a web browser window with the address bar containing '127.0.0.1:5000/nnwtablestates/5'. The page header features the 'Gibbs system' logo and navigation links: 'Главная', 'Список элементов', 'Регистрация элемента', and 'Сост'. The main content area is titled 'Регистрация состояний' and contains the following text: 'Обнаружено 5 состояний', 'Введите имена состояний через запятую согласно обучающему набору данных', and 'Названия состояний: 1,2,3,4,5'. Below the text is a button labeled 'Занести элемент'. At the bottom of the page, there is a footer with the text '© ООО СТЭЙТИНТЕЛ, 2022'.

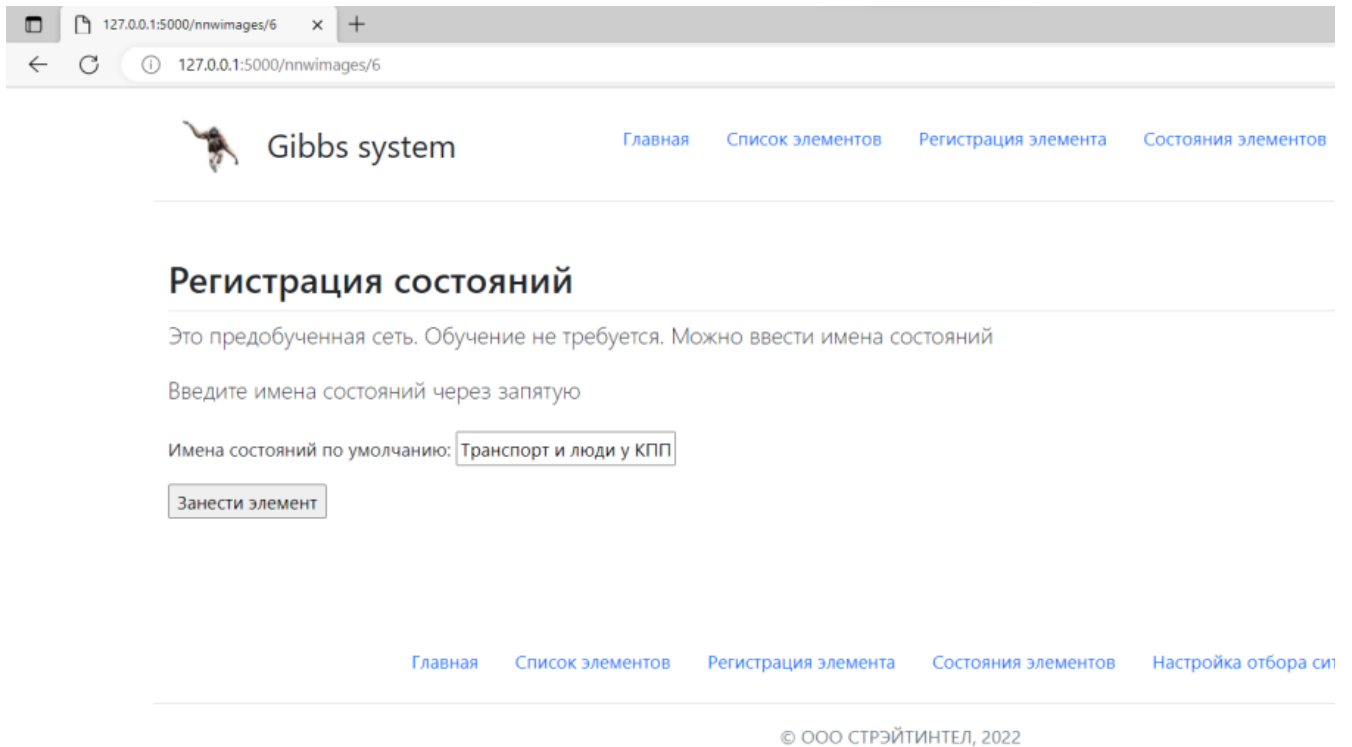
После обучения система укажет, сколько обнаружено классов на основе обучающего набора данных, и предложит внести имена для всех классов через запятую.

В случае пропуска шага, имена состояний примут числовой порядок $1,2,3\dots n$, где n – количество обнаруженных классов.

3.5. Обучение классификатора на основе машинного зрения «Нейросеть - обнаружение людей и транспорта»

Предобученная сеть, не требующая обучения.

При переходе по ссылке «Настроить классификатор» откроется вкладка для регистрации имен состояний. Где будет предложено внести имена состояниям.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/nnwimages/6'. The page header includes the 'Gibbs system' logo and navigation links: 'Главная', 'Список элементов', 'Регистрация элемента', and 'Состояния элементов'. The main content area is titled 'Регистрация состояний' and contains the following text: 'Это предобученная сеть. Обучение не требуется. Можно ввести имена состояний', 'Введите имена состояний через запятую', and 'Имена состояний по умолчанию: Транспорт и люди у КПП'. Below this is a text input field containing 'Транспорт и люди у КПП' and a button labeled 'Занести элемент'. A secondary navigation bar at the bottom of the page includes links for 'Главная', 'Список элементов', 'Регистрация элемента', 'Состояния элементов', and 'Настройка отбора си'. The footer contains the copyright notice '© ООО СТЭЙТИНТЕЛ, 2022'.

3.6. Обучение классификатора на основе машинного зрения «Нейросеть - обнаружение объекта»

Предобученная сеть, не требующая обучения.

При переходе по ссылке «Настроить классификатор» откроется вкладка для выбора элементов для обнаружения.

Настройка обнаружения

127.0.0.1:5000/nnwimages1/6

Gibbs system

[Главная](#) [Список элементов](#) [Регистрация эле...](#)

Имена

Выберите тип элемента:

- люди
- рюкзак
- сумка
- кейс

[Занести элемент](#)

[Главная](#) [Список элементов](#) [Регистрация элемента](#) [Состояния эле...](#)

© ООО СТЭЙТИНТЕЛ, 2022

Необходимо из списка выбрать элемент или несколько элементов (до 10 из каталога), при обнаружении которых на изображении классификатор будет выдавать сигнал обнаружения.

Классификатор выдает сигнал при обнаружении хотя бы одного элемента из выбранного списка.

3.7. Обучение классификатора на основе машинного зрения «Обнаружение изменения на территории»

При переходе по ссылке «Настроить классификатор» откроется вкладка для настройки классификатора.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/nnwimages2/6'. The page header includes the 'Gibbs system' logo and navigation links: 'Главная', 'Список элементов', 'Регистрация элемента', and 'Состояния эле'. The main heading is 'Настройка машинного зрения для элемента: камера 1 - въезд'. Below this, there are two input fields: 'Время переобучения: 30' and 'Площадь изменений: 30'. A 'Сохранить настройки' button is located below the second input field. At the bottom of the page, there is a copyright notice: '© ООО СТЭЙТИНТЕЛ, 2022'.

1) Необходимо выбрать время переобучения классификатора – время через которое будет обновляться фон – получение среднего кадра без динамических объектов, с которым сравнивается актуальный кадр.

2) Необходимо ввести процент изменений – размер области по отношению ко всему размеру кадра, при изменении которой выдается сигнал. Например, необходимо выдавать сигнал, если на кадре произошли изменения целой области в отношении 30% (целой области размером 30% от площади всего кадра – не суммарно 30% малых фрагментов) от площади всего кадра – необходимо записать 30.

3) Необходимо сохранить настройки.

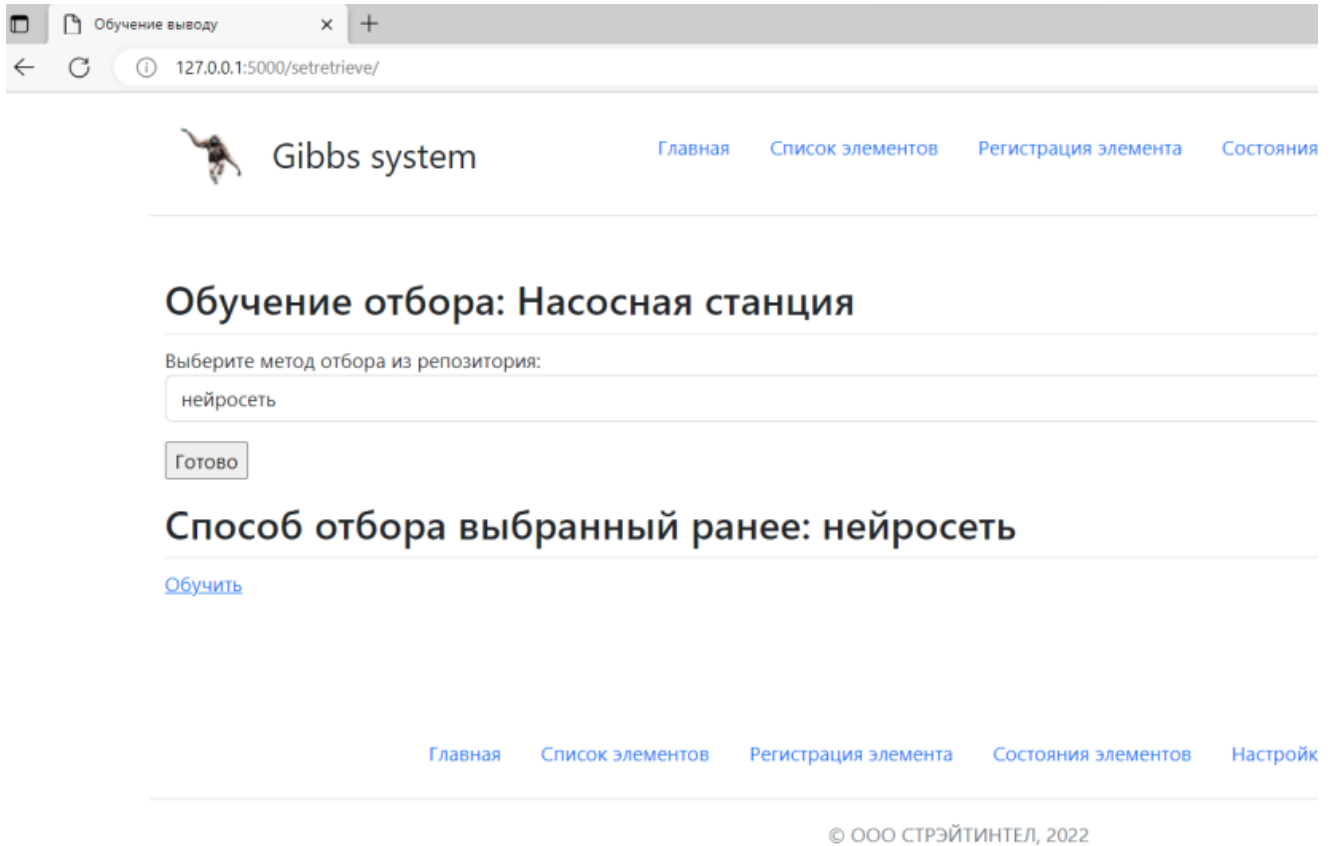
3.8. Обучение классификатора на основе машинного зрения «Нейросеть - классификация погодных условий»

Предобученная сеть, не требующая обучения.

При переходе по ссылке «Настроить классификатор» откроется вкладка для регистрации имен состояний. Где будет предложено внести пользовательские имена состояниям.

4. ОБУЧЕНИЕ ОТБОРА СИТУАЦИИ ИЗ БАЗЫ

1) Перейти на вкладку настройка отбора ситуаций



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/setretrieve/'. The page header includes the Gibbs system logo and navigation links: Главная, Список элементов, Регистрация элемента, and Состояния. The main content area is titled 'Обучение отбора: Насосная станция' and contains a form with the label 'Выберите метод отбора из репозитория:'. A dropdown menu is open, showing 'нейросеть' as the selected option. Below the dropdown is a 'Готово' button. Underneath, the text 'Способ отбора выбранный ранее: нейросеть' is displayed, followed by a blue 'Обучить' link. At the bottom of the page, there are additional navigation links: Главная, Список элементов, Регистрация элемента, Состояния элементов, and Настройк, along with a copyright notice: © ООО СТЭЙТИНТЕЛ, 2022.

2) Выбрать метод отбора из списка

3) Нажать кнопку готово

4) Перейти по ссылке «Обучить»

Если выбрана нейросеть:

5) Будет продемонстрирован вектор ситуации (конкатенация векторов от всех настроенных классификаторов) для проверки соответствия результата работы системы обучающему набору данных.

В обучающем наборе должны быть такие же вектора (пара сравниваемых ситуаций в виде векторов), последний символ – процент их схожести.

6) Необходимо загрузить обучающий набор данных и нажать кнопку «Запустить обучение».

7) По окончании успешного обучения откроется список всех состояний всех элементов. Необходимо выбрать нежелательные состояния, при которых будет происходить процедура отбора ситуаций из базы и вывод решения.

8) Нажать кнопку «Загрузить».

Если выбрана аддитивная свертка:

5) Будет продемонстрирован вектор ситуации (конкатенация векторов от всех настроенных классификаторов). Будет предложено внести веса относительной важности элементов.

6) Нажать кнопку «Загрузить».

7) Откроется список всех состояний всех элементов. Необходимо выбрать нежелательные состояния, при которых будет происходить процедура отбора ситуаций из базы и вывод решения.

8) Нажать кнопку «Загрузить».

5. ОПИСАНИЕ МОДУЛЕЙ КЛАССИФИКАЦИИ СОСТОЯНИЙ

Классификация состояний элементов осуществляется в подключаемых классификаторах.

Для анализа входных данных, поступающих от элементов сложного технологического объекта, и их преобразования в вектор S необходим соответствующий классификатор, который должен быть адаптирован к типу входной информации.

Классификация состояния элемента может основываться на данных, получаемых в реальном времени с датчиков системы мониторинга и телеметрии, включая как количественные, так и качественные показатели. Такой подход характерен для технологического оборудования, а также для интерфейсов и линий связи, где данные передаются непосредственно.

Кроме того, классификация может осуществляться на основе данных, полученных в результате анализа косвенных признаков или другой качественной информации. Это особенно актуально для систем операционного окружения, где непосредственные данные не собираются в реальном времени, но могут быть доступны по запросу или через обращения к сторонним базам данных корпоративных информационных систем. К таким данным могут относиться информация о состоянии сервисных служб и персонала, наличие ресурсов, объекты инфраструктуры и другие важные аспекты.

5.1. Модуль идентификации состояния объекта мониторинга на основе экспертных правил

Состоит из трех основных частей: таблица базы правил в формате SQLite, функция занесения экспертных правил, функция идентификации состояния объекта мониторинга (ОМ).

Функциональные возможности: ручное занесение экспертных правил в базу правил, интерпретация параметров ОМ (значения параметров и/или логические значения) и классификация актуального состояния ОМ на основе экспертных правил. Результат работы – вектор $[x_1, x_2, \dots, x_n]$, где каждая позиция x принимает

значение в диапазоне [0:1] и означает вероятность нахождения ОМ в определенном состоянии.

Алгоритм работы:

1. При настройке пользователь вносит в базу правил с помощью формы ввода правила классификации элемента в формате «состояние - параметры - значения - условие».
2. Модуль считывает актуальные параметры из источника данных.
3. Модуль выдает вектор S , описывающий состояние элемента, заносит актуальное состояние в БД.

Возможна нечеткая классификация в случае отсутствия уверенности в том или ином состоянии. Рассмотрим подробнее на упрощенном примере теплового насоса, который описывается 3 атрибутами (температура T , давление P , автоматика двигателя A) и имеет 4 состояния:

Имя состояния	Символы параметров	Значения параметров (диапазоны)	Условия принадлежности
Работает	T,P,A	40-80,150-200,ON	1
Остановлено	T,P,A	20-80,50-200,OFF	1
Авария	T,P	>80,>200	0
Прогрев	T,P,A	0-40,0-100,ON	1

1 и 0 обозначают условия принадлежности.

1 – объект мониторинга принимает состояние «исправно» только если все актуальные параметры совпадают с условиями (указанными значениями). 0 – объект мониторинга принимает состояние «Авария», если хотя бы один параметр в актуальной ситуации принял значение согласно правилу (например, $T=90$).

5.2. Модуль классификации состояния элемента на основе экспертных правил для проверки логических значений

Состав модуля: функция идентификации состояния элемента, функция настройки имен состояний. Дополнительно в состав включена функция-обработчик ошибок, проверяющая доступ к БД, наличие элемента в БД конфигурации прикладной системы, источник данных и их формат.

Основная функция: классификация состояния элемента на основе логических правил: Функция: классификация состояния элемента True/False на основе единичного параметра.

5.3. Модуль классификации состояния элемента на основе нейронных сетей для нормированных данных

Состав модуля: интерфейс для загрузки обучающего и проверочного набора данных, функция обучения, включающая необходимые слои архитектуры, функция применения нейросети. Дополнительно в состав включена функция-обработчик ошибок.

Модуль универсален, подстраивается под различные размеры обучающего набора данных (под разное количество атрибутов).

Для каждого элемента со своим обучающим набором применяется типовая функция обучения. Функция применения срабатывает с персональной для элемента конфигурацией модели, созданной в результате обучения классификатора на этапе сборки прикладной системы. Конфигурация хранится в папке директории Платформы с именем, соответствующим id элемента. Конфигурация модели содержит веса и конфигурацию обученной нейросети, имена состояний.

Модель классификатора ансамблевая, состоит из двух одновременно обучающихся сетей. За основу архитектуры модели взята полносвязная нейронная сеть для многоклассовой классификации (Функция активации выходного слоя Softmax, функция ошибки sparse_categorical_crossentropy, метрика accuracy, optimizer – adam, количество нейронов на выходе нейросети вариативно и равно

количеству обнаруженных классов). Для ансамблирования применен слой `keras.tf concatenate`.

Входные данные: нормированные по категориям количественные параметры, отнесенные к диапазонам, качественные параметры (логические, принимающие значение только 0/1).

5.4. Модуль классификации состояния элемента на основе нейронных сетей для ненормированных данных

Модуль универсален, подстраивается под различные размеры обучающего набора данных (под разное количество атрибутов).

Для каждого элемента со своим обучающим набором применяется типовая функция обучения. Функция применения срабатывает с персональной для элемента конфигурацией модели, созданной в результате обучения классификатора на этапе сборки прикладной системы. Конфигурация хранится в папке директории Платформы с именем, соответствующим `id` элемента. Конфигурация модели содержит веса и конфигурацию обученной нейросети, имена состояний.

Модель классификатора ансамблевая, состоит из четырех одновременно обучающихся сетей. За основу архитектуры модели взята полносвязная нейронная сеть для многоклассовой классификации (Функция активации выходного слоя `Softmax`, функция ошибки `sparse_categorical_crossentropy`, метрика `accuracy`, `optimizer – adam`, количество нейронов на выходе нейросети вариативно и равно количеству обнаруженных классов). Для ансамблирования применен слой `keras.tf concatenate`.

5.5. Модуль классификации состояния элемента на основе компьютерного зрения «Нейросеть - обнаружение людей и транспорта».

Модуль работает с изображением. Модуль использует предобученную модель нейронной сети YOLO. В данном модуле предусмотрена реакция на

людей и транспорт (легковые автомобили, грузовые, автобусы, мотоциклы, велосипеды).

Входные данные: изображение с камеры наблюдения.

Результат работы – вектор S , характеризующий состояние.

Длина вектора равна 4 позициям:

[1,0,0,0] – «Пусто»;

[0,1,0,0] – «Прибыл транспорт»;

[0,0,1,0] – «Транспорт и люди обнаружены»;

[0,0,0,1] – «Люди обнаружены».

Дополнительно предусмотрена настройка имен перечисленных состояний.

5.6. Модуль классификации состояния элемента на основе компьютерного зрения «Нейросеть - обнаружение различных объектов»

Модуль использует предобученную модель нейронной сети YOLO. Модель способна распознавать 63 различных объектов на изображении.

В данном модуле предусмотрена настройка – выбор объектов из списка (до 10 элементов). При обнаружении выбранных объектов на изображении классификатор будет выдавать сигнал обнаружения. Классификатор выдает сигнал при обнаружении хотя бы одного элемента из выбранного списка.

Потенциально возможно дообучение классификатора на обнаружение пользовательских объектов.

Входные данные: изображение с камеры наблюдения.

5.7. Модуль классификации состояния элемента на основе компьютерного зрения «Обнаружение изменения на территории»

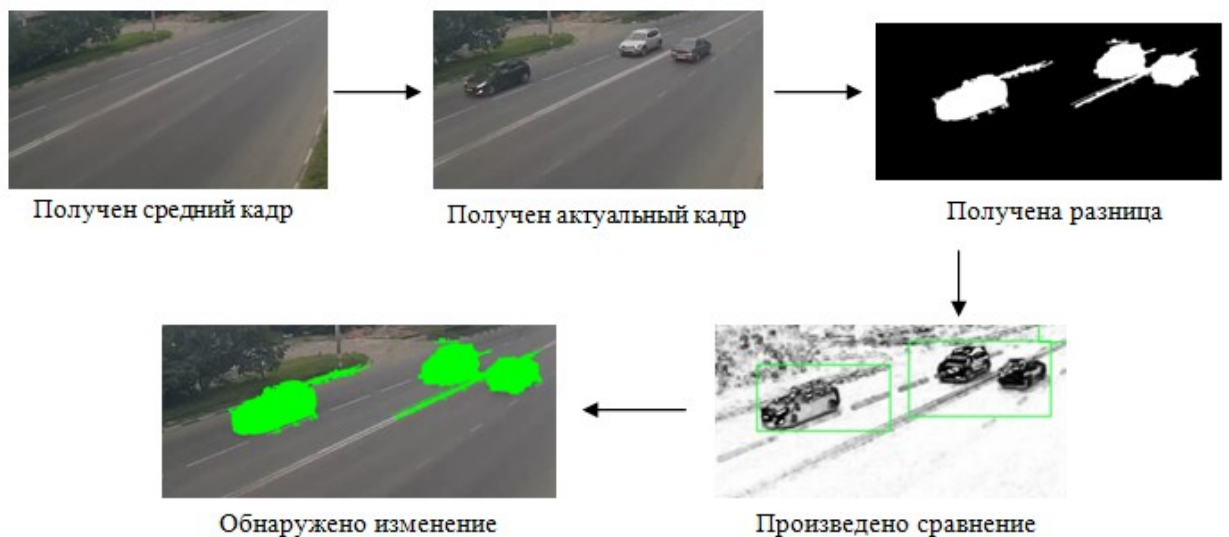
Модуль на основе компьютерного зрения, использует библиотеку компьютерного зрения OpenCV. Основная функция модуля: обнаружение изменений на кадре без идентификации обнаруженных изменений.

Целесообразно использовать данный функционал для мониторинга помещений и территорий на предмет появления посторонних различных

объектов, перемещений различных объектов, появления различных препятствий на путях движения транспорта и людей, деформаций и изменений конструкций. Модуль не требует длительного обучения и большого объема данных.

Принцип работы: актуальный кадр сравнивается с заданным кадром, принятым за образец с помощью функции `structural_similarity` библиотеки `skimage`. Если изменения превышают заданный порог площади изображения, то выдается сигнал о наличии изменения.

Принцип работы в рисунках представлен на следующем рисунке: «Обнаружение изменения на территории»



Кадр, принятый за образец, может быть задан вручную или через функцию получения медианного кадра (средний кадр из видео, где отсутствуют постоянно движущиеся в кадре объекты).

Дополнительно предусмотрена настройка допустимой площади изменения.

Входные данные: изображение с камеры наблюдения.

Результат работы – вектор S , характеризующий состояние: $[0,1]$ – не обнаружено, $[1,0]$ – изменения обнаружены.

5.8. Модуль классификации состояния элемента на основе компьютерного зрения «Нейросеть - классификация погодных условий»

Предобученная нейронная сеть на основе машинного зрения. Способна классифицировать погодные условия и их эффекты влияния на основе изображений. За основу сети взята предобученная сверточная нейронная сеть VGG16 для выделения базовых признаков изображения.

Сеть классифицирует погодные условия: снегопад, дождь, молнии, наледь. Не требует обучения на этапе сборки системы, необходимая конфигурация уже загружена в платформу.

Модель целесообразно также применять для обнаружения луж, сугробов, наледи на конструкциях.

Входные данные: изображение с камеры наблюдения.

Результат работы – вектор S , характеризующий состояние длиной 4 элемента.

6. СИТУАЦИОННАЯ БАЗА ЗНАНИЙ

Выходной реакцией ИСППР является аргументированное решение, содержащееся в прецеденте знаний или сгенерированное на основе преобразований прецедентов в базе знаний. Аргументация основывается на аналогии, когда пользователю объясняется: где, когда, кем, в каких условиях применялось (рекомендовалось) аналогичное решение.

Решение - инструкция по переводу объекта из критической ситуации в целевую исправную. Решение может быть адресовано конечным пользователям.

Каждому конечному пользователю предназначается персональная информация. Это может быть непосредственно технологическая карта действий, справочный материал, контакты и т.п. Общая структура решения представлена следующими компонентами:

- D1 – инструкция по действиям в ситуации (технологическая карта);
- D2 – список, контакты ответственных лиц и необходимых организаций;
- D3 – требуемые отчетные документы (шаблоны, бланки, акты и т.п.);
- D4 – дополнительная справочная информация (ссылка на аналогичные ситуации, экспертные рекомендации и т.п.).

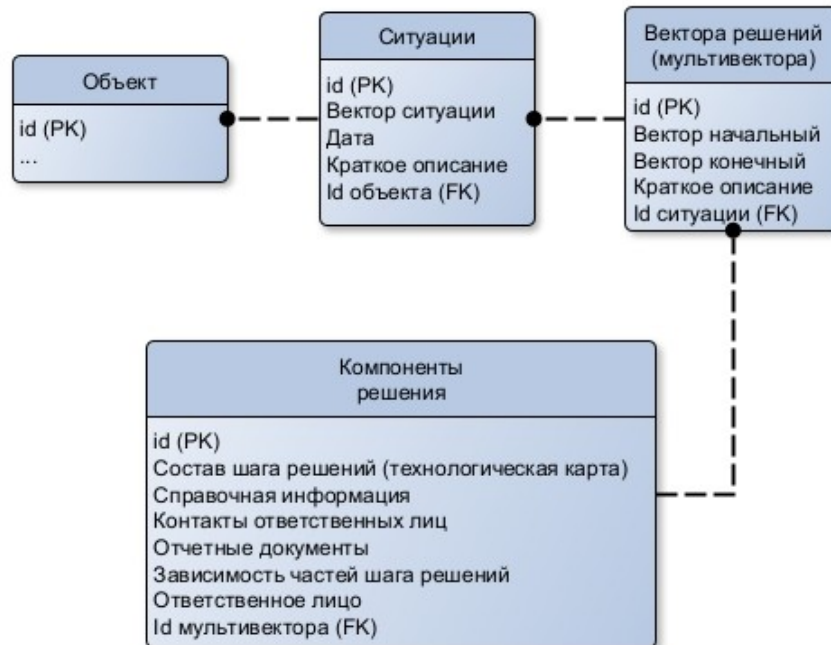
Каждый шаг из последовательности $R^\gamma = (\check{S}^0, \check{S}^3, \dots, \check{S}^k, \dots, \check{S}^{k_{\text{end}}})$ содержит в себе свой набор компонентов D.

Ситуационная база знаний состоит из трех таблиц, содержит прецеденты:

- таблица с векторами \check{S} , которые выступают в роли ключа для отбора ситуации;
- таблица мультивекторов R^γ (формальное представление решения Sol некоторой ситуации Sit), связанных с ключом \check{S} ;

- таблица компонентов решения D, соответствующих каждому шагу из последовательности R^y .

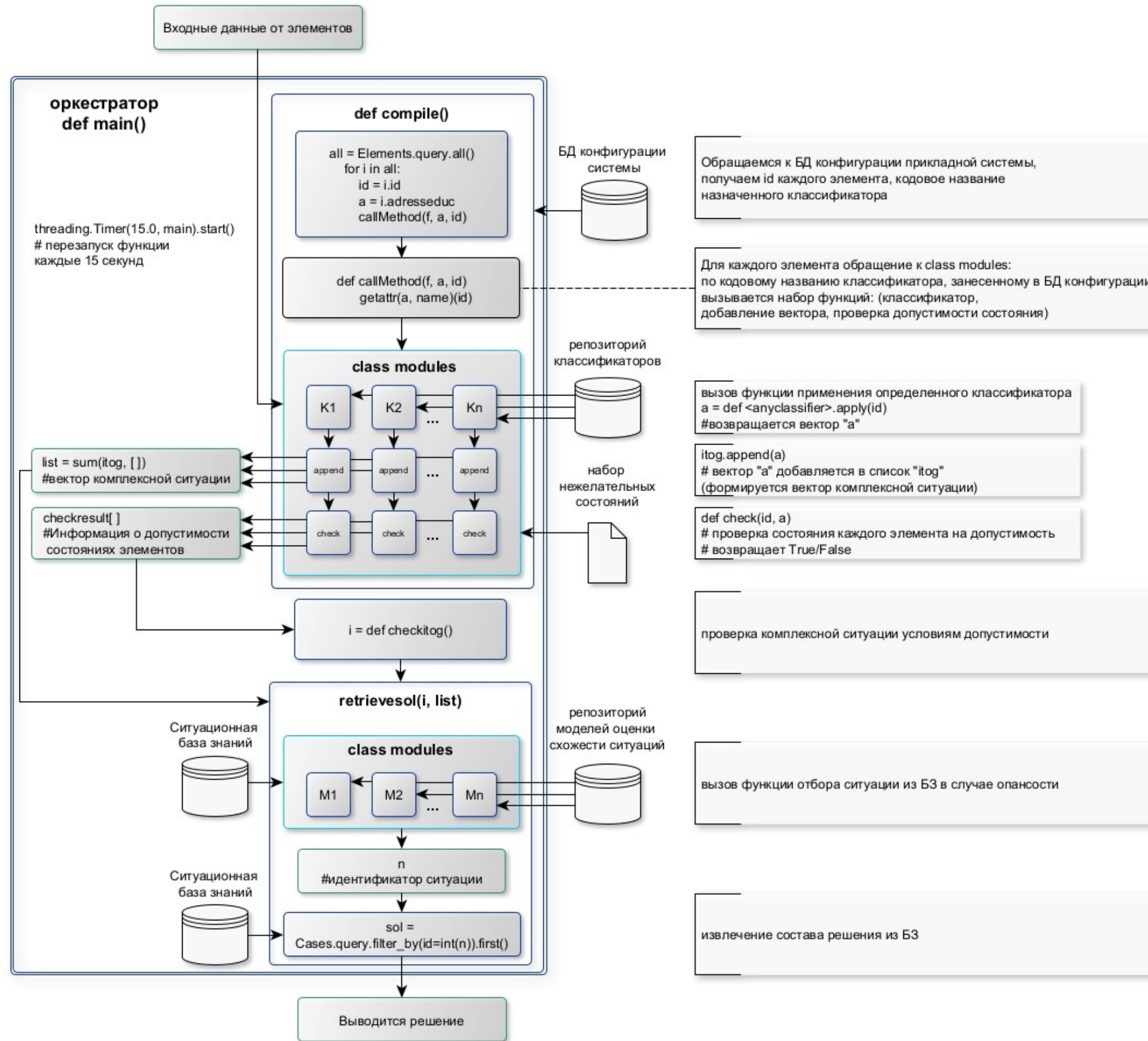
Логическая модель данных БЗ представлена на следующем рисунке:



БД реализована с помощью SQLite. Взаимодействие осуществляется через модуль SQLAlchemy библиотеки flask_sqlalchemy.

Приложение 2

Модуль оркестратор (def compile)



Физическая архитектура модуля «Оркестратор» (def main())

Листинг функции Оркестратора

```
from kbs import Elements
from kbs import Cases, Object
import os
from admin import rules
from admin import nnwtable
from admin.imageoperation import yolo
from admin import logic
from admin import retrievetrain
import threading
from kbs import UPLOAD_FOLDER
import json
IMAGEED_FOLDER = os.path.join('admin', 'imageoperation')

itog = []
checkres = []
num = []

class modules:
    def rules(self, id):
        id = id
        a = rules.rulesappl(id)
        print(a,id)
        #itog[id]=a
        itog.append(a)
        check(id, a)
        return
    def nnwtable(self, id):
        id = id
        a = nnwtable.nnwtableapply(id)
        print(a,id)
        itog.append(a)
        check(id, a)
        return
    def nnwimages(self, id):
        id = id
```

```

        a = yolo.nnwimages(id)
        #print(a)
        print(a, id)
        itog.append(a)
        check(id, a)
        return
def logic(self, id):
    id = id
    a = logic.logic(id)
    print(a, id)
    itog.append(a)
    check(id, a)
    return
def retrnnw1(self, list):
    num.append(retrievetrain.apply(list))
    print(num)
    return

def callMethod(f, name, id):
    getattr(f, name)(id)

def callMethod1(f, name, list):
    getattr(f, name)(list)

f = modules()
#callMethod(f, "rul")
def compile():
    all = Elements.query.all()
    for i in all:
        id = i.id
        name = i.adresseduc
        callMethod(f, name, id)

def main():
    threading.Timer(10.0, main).start()
    ob = Object.query.get_or_404(1)
    itog.clear()
    checkres.clear()
    num.clear()

```



```

if ob.adressmodel == "1": #если запущено распознавание
    compile() #опрос состояний, добавление состояний в
итоговый список
    list = sum(itog, []) #формирование вектора
комплексной ситуации путем объединения итогового списка
    print(list)

    #print(checkres)
    i = checkitog() #проверка комплексной ситуации на
опасность
    if ob.adressappl == "1": #если запущен отбор
        s = retrievesol(i, list) #определение схожести
ситуации, вывод решения в случае обнаружения опасности
        return s
    else:
        print("отбор отключен")
        return "отбор отключен"

else:
    print("классификация отключена")
    return "классификация отключена"

def checkitog():
    if False in checkres: #в случае наличия недопустимых
состояний возвращается False
        return False
    else:
        return True #в случае отсутствия недопустимых
состояний возвращается True

def retrievesol(i, list):
    if i == False:
        ob = Object.query.get_or_404(1)
        a = ob.adresseduc
        callMethod1(f, a, list)
        #print(num)
        n = num[0]
        sol = Cases.query.filter_by(id=int(n)).first()
        #print(sol.description)

```

```
        s = sol.description
else:
    s = ("Все в порядке")
print(s)
return s
```

```
def check(id, a):
    j=[]
    try:
        with open(f"{UPLOAD_FOLDER}\\main.json",
encoding="utf8") as f:
            file_content = f.read()
            templates = json.loads(file_content)
            #print(templates[f"{id}"])
            ex = templates[f"{id}"]
            len(ex)
            for i in range(len(ex)):
                j.append(list(ex[i].values())[0])
            #print(j)
        if a in j:
            #print(f"{id} допустимо")
            checkres.append(True)
            return True
        else:
            #print(f"{id} не допустимо")
            checkres.append(False)
            return False
    except:
        pass
```

```
main()
```

Приложение 3

Свидетельства о регистрации РИД

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022680042

Платформа для быстрого создания интеллектуальных систем поддержки принятия решений "ГИББС-СИСТЕМ"

Правообладатель: **ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "СТРЭЙТИНТЕЛ" (RU)**

Авторы: **Глухих Дмитрий Игоревич (RU), Глухих Игорь Николаевич (RU)**

Заявка № **2022669294**

Дата поступления **19 октября 2022 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **26 октября 2022 г.**



*Руководитель Федеральной службы
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b80077b14c1210a94edbd24145d5c7
Владелец **Зубов Юлий Сергеевич**
Действителен с 26.10.2022 по 26.05.2023

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022614358

**Модуль идентификации состояния объекта
мониторинга на основе экспертных правил**

Правообладатель: **ОБЩЕСТВО С ОГРАНИЧЕННОЙ
ОТВЕТСТВЕННОСТЬЮ "СТРЭЙТИНТЕЛ" (RU)**

Автор(ы): **Глухих Дмитрий Игоревич (RU)**



Заявка № **2022613116**

Дата поступления **01 марта 2022 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **21 марта 2022 г.**

*Руководитель Федеральной службы
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b90077e14e1703d94e6dc24145d9c7
Владелец **Зубов Юрий Сергеевич**
Действителен с 20.05.22 по 26.05.2023

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**RU2024612231**

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства): 2024612231 Дата регистрации: 30.01.2024 Номер и дата поступления заявки: 2023689437 27.12.2023 Дата публикации и номер бюллетеня: 30.01.2024 Бюл. № 2 Контактные реквизиты: a.p.volosnikov@utmn.ru	Автор(ы): Глухих Игорь Николаевич (RU), Глухих Дмитрий Игоревич (RU), Хисамутдинова Гульноз (ТJ), Прохошин Алексей Сергеевич (RU) Правообладатель(и): ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ" (RU)
---	--

Название программы для ЭВМ:
Программный модуль системы поддержки принятия решений на объектах фитосанитарного мониторинга

Реферат:

Для вывода альтернатив решений по мероприятиям при выращивании продовольственных агрокультур в тепличных хозяйствах и сити-фермах с учетом экономических и экологических факторов. Позволяет повысить производственную и экономическую эффективность тепличных хозяйств и сити-ферм, снизить требования к квалификации сотрудников. Используется в эксплуатации и управлении сложными технологическими объектами. Функциональные возможности: Прием и обработка входных данных о состояниях контролируемых параметров тепличного хозяйства; Вывод решения по входным данным о ситуации; Классификация состояний контролируемых параметров тепличного хозяйства по входным данным; Хранение, редактирование, удаление, добавление технологических карт; Хранение, редактирование, удаление, добавление сведений о решениях; Хранение, редактирование, удаление, добавление сведений о ситуациях. Тип ЭВМ: PC 7th Gen Intel(R) Core(TM) i3 и выше. 2ГБ ОЗУ и выше. ОС: Windows 10 и выше.

Язык программирования: Java, Python, Языки разметки: html, css

Объем программы для ЭВМ: 1,9 ГБ

Приложение 4

Акт внедрения



SMART-РОСТ

Акт внедрения

Материалов диссертационной работы Глухих Дмитрия Игоревича,
Представленной на соискание ученой степени кандидата технических наук

Настоящим актом подтверждается, что результаты диссертационной работы Д.И. Глухих внедрены в систему «СМАРТ-РОСТ». Метод математического моделирования ситуации на сложном технологическом объекте, метод моделирования процесса выработки решения на базе гибридного метода рассуждения на основе прецедентов используются в системе поддержки принятия решения на объектах агрокомплекса и частных хозяйствах. Программное обеспечение использует предложенные Глухих Д.И. методы вместе с большими языковыми моделями и технологиями retrieval augmented generation (RAG) в целях вывода пользователю решения: как лечить растение, как поддерживать растение для лучшего урожая. Система реализуется в рамках проекта «Программная платформа с применением нейросетей компьютерного зрения и больших языковых моделей для поддержки принятия решений на объектах агрокомплекса» (Договор 430ГС1ЦТС10-D5/101544 от 11.11.2024, Фонд содействия инновациям).

Генеральный директор
ООО «СМАРТ-РОСТ»

Милос В.Н. Щербакова

