



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР «КУРЧАТОВСКИЙ ИНСТИТУТ»



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
«ФЕДЕРАЛЬНЫЙ НАУЧНЫЙ ЦЕНТР НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ СИСТЕМНЫХ ИССЛЕДОВАНИЙ
НАЦИОНАЛЬНОГО ИССЛЕДОВАТЕЛЬСКОГО ЦЕНТРА «КУРЧАТОВСКИЙ ИНСТИТУТ»
(НИЦ «КУРЧАТОВСКИЙ ИНСТИТУТ» - НИИСИ)

На правах рукописи

Мащенко Кирилл Алексеевич

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ МОДУЛЕЙ
ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ПЛАТФОРМ

2.3.8. Информатика и информационные процессы

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
кандидат физико-математических наук, доктор педагогических наук, доцент
Леонов Александр Георгиевич

Москва – 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
ГЛАВА 1. АНАЛИЗ ЭВОЛЮЦИИ И ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ПЛАТФОРМ С РАЗЛИЧНЫМ УРОВНЕМ АВТОМАТИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА	14
1.1 Эволюция автоматизированных и интеллектуальных обучающих систем	15
1.2 Сравнительный анализ современных цифровых образовательных платформ и интеллектуальных модулей автоматизации образовательного процесса.....	17
1.3 Ограничения существующих решений и обоснование необходимости разработки новой отечественной цифровой образовательной платформы с универсальными интеллектуальными модулями	28
Выводы по главе 1	29
ГЛАВА 2. РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ МОДУЛЕЙ ДЛЯ АВТОМАТИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА, В ТОМ ЧИСЛЕ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА	32
2.1 Модуль учёта посещаемости студентов в условиях очного, дистанционного и гибридного обучения	32
2.1.1 Анализ существующих подходов к автоматизированному учёту посещаемости.....	32
2.1.2 Алгоритмы учёта посещаемости при очных форматах обучения	34
2.1.3 Алгоритмы учёта посещаемости при дистанционных форматах обучения	36
2.1.4 Интегрированная архитектура учёта гибридной посещаемости	37
2.1.5 Контрольные метрики и верификация корректности учёта посещаемости по фото	42

2.2 Модуль обнаружения заимствований в студенческих решениях	44
2.2.1 Анализ существующих подходов по выявлению заимствований в образовательных системах	44
2.2.2 Алгоритм обнаружения заимствований, учитывающий синтаксические особенности языков программирования	46
2.2.3 Использование модуля обнаружения заимствований в различных образовательных сценариях	50
2.3 Модуль семантической проверки свободных текстовых ответов обучающихся.....	53
2.3.1 Анализ существующих методов автоматической проверки открытых текстовых ответов.....	54
2.3.2 Лингвистический процессор и построение графов синтаксического разбора	57
2.3.3 Предобработка синтаксических графов и обработка отрицаний	60
2.3.4 Сравнение синтаксических деревьев и вычисление итоговой метрики.....	62
2.4 Нейросетевые ассистенты на основе контекстно-ограниченного взаимодействия в образовательном процессе.....	68
2.4.1 Анализ существующих подходов к разработке интеллектуальных диалоговых систем	68
2.4.2 Предметно-ориентированный чат-бот с заданной областью знаний	70
2.4.3 Использование больших языковых моделей в образовательных системах.....	71
2.4.4 Нейросетевой ассистент обучающегося для контекстной индивидуальной поддержки.....	72
2.4.5 Нейросетевой ассистент преподавателя для создания образовательного контента и сопровождения занятий.....	79
Выводы по главе 2	80

ГЛАВА 3. МАТЕМАТИЧЕСКИЕ МЕТОДЫ И АЛГОРИТМЫ ДЛЯ ЗАДАЧ АНАЛИЗА ЦИФРОВОГО СЛЕДА ОБУЧАЮЩИХСЯ И УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ.....	82
3.1 Анализ существующих методов сбора и анализа цифрового следа в образовательных системах	82
3.2 Модуль сбора и отображения аналитики данных цифрового следа	85
3.3 Сигнатурные методы анализа временных рядов.....	92
3.3.1 Математический аппарат сигнатур путей.....	95
3.3.2 Логарифмические сигнатуры и их свойства	99
3.4 Генерация синтетических данных, сохраняющих статистические свойства исходных траекторий	101
3.4.1 Обращение логарифмической сигнатуры, теорема об оценке погрешности.....	103
3.4.2 Эволюционный алгоритм обращения логарифмической сигнатуры	105
Выводы по главе 3	108
ГЛАВА 4. ИНТЕГРАЦИЯ И АПРОБАЦИЯ РЕАЛИЗОВАННЫХ МОДУЛЕЙ НА БАЗЕ ЦИФРОВОЙ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ МИРЕРА	110
4.1 Архитектура платформы и механизмы интеграции разработанных интеллектуальных модулей	111
4.2 Архитектурные подходы к интеграции цифровых образовательных сред	115
4.2.1 Интеграция цифровой образовательной среды ЭВМ-практикум	117
4.2.2 Интеграция цифровой образовательной среды КуМир.....	119
4.2.3 Интеграция цифровой образовательной среды ПиктоМир.....	120
4.2.4 Интеграция цифровой образовательной среды Электронные таблицы.....	122

4.2.5 Распределенная архитектура интеграции с внешними образовательными платформами.....	123
4.3 Экспериментальная апробация и оценка эффективности разработанных модулей в условиях цифровой образовательной платформы	125
4.3.1 Оценка эффективности модуля автоматизированного учёта посещаемости.....	126
4.3.2 Оценка эффективности модуля обнаружения заимствований в программных решениях	128
4.3.3 Оценка эффективности модуля автоматизированной семантической проверки текстовых ответов	130
4.3.4 Оценка эффективности применения нейросетевых ассистентов в образовательном процессе.....	131
4.3.5 Оценка эффективности модуля аналитики цифрового следа обучающихся	135
4.3.6 Оценка эффективности сигнатурных методов при анализе и прогнозировании образовательных траекторий.....	136
Выводы по главе 4	140
ЗАКЛЮЧЕНИЕ.....	142
СПИСОК ЛИТЕРАТУРЫ	144
ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВА О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ.....	162
ПРИЛОЖЕНИЕ Б. АКТЫ О ВНЕДРЕНИИ В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС	164

ВВЕДЕНИЕ

Актуальность темы исследования

Цифровая экономика ставит задачу массовой подготовки высококвалифицированных специалистов, особенно в области информационных технологий и инженерно-технических дисциплин. При этом цифровая трансформация образования пока не обеспечивает необходимого уровня автоматизации учебного процесса: на преподавателях лежит значительная рутинная нагрузка, связанная с проверкой заданий, ведением отчетности, организацией очного и дистанционного обучения и контролем посещаемости. Это ограничивает возможности индивидуализации обучения, увеличения объема практических заданий и предоставления персонализированной обратной связи. Ключевой задачей становится разработка платформ и интеллектуальных модулей, автоматизирующих рутинные действия преподавателей и расширяющих инструменты содержательной педагогической деятельности. Актуальность подтверждается активизацией исследований по применению искусственного интеллекта в образовании. Задача разработки масштабируемой интеграционной платформы для комбинированного очного и дистанционного обучения с высоким уровнем автоматизации образовательного процесса в вузах ни в России, ни в мире в полной мере не решена. В связи с этим актуальность темы исследования определяется необходимостью разработки универсальных интеллектуальных модулей, обеспечивающих высокий уровень автоматизации образовательного процесса и интегрируемых в цифровые образовательные платформы.

Разработанность темы исследования

Исследования в области автоматизации образовательного процесса связаны с интеллектуальными обучающими системами, предложившими адаптивное обучение, проверку заданий, анализ действий обучающихся и управление обучением. Одной из первых стала SCHOLAR, где обучение строилось как диалог на основе семантических сетей и анализа ошибок обучаемых. Затем получили развитие системы управления обучением (LMS, англ. Learning Management System)

Moodle, Blackboard, Google Classroom и отечественные Modeus, In Study и Эквио, реализующие управление курсами, оценку знаний, коммуникацию, дистанционное и смешанное обучение. Масштабируемое онлайн-обучение обеспечивают платформы массовых открытых онлайн-курсов Coursera, Stepik, Открытое образование и Универсариум. В школьном образовании используются СберКласс и Якласс с элементами интеллектуальной поддержки. Методы искусственного интеллекта, учебной аналитики и анализа цифрового следа реализуют AutoTutor, WatsonTutor, CourseSignals и Пифия для автоматизации обратной связи, персонализации и прогнозирования образовательных результатов. Проверку задач по программированию автоматизируют ejudge, Яндекс.Контест и Codeforces.

Значительный вклад в формирование отечественных образовательных сред внесли отечественные исследователи А.П. Ершов (теория программирования и школьная информатика), В.Б. Бетелин (системное программное обеспечение и программные комплексы), А.Л. Семёнов (информатизация образования и методика обучения информатике), А.Г. Кушниренко (учебные программные среды и алгоритмическое обучение), А.Г. Гейн (содержание и учебники школьной информатики), С.Г. Григорьев (ИКТ в образовании), Ю.А. Первин (методика преподавания информатики), В.Г. Житомирский (учебно-методическое обеспечение информатики), Я.Н. Зайдельман (учебные материалы и курсы по информатике), Г.А. Звенигородский (структурирование содержания школьной информатики), А.А. Кузнецов (образовательные программы и стандарты по информатике), Г.В. Лебедев (учебно-методические комплексы и алгоритмическое мышление), А.Г. Леонов (программные средства поддержки обучения).

При этом существующие решения преимущественно автоматизируют отдельные задачи или базовые функции управления обучением. Интеллектуальные модули для проверки заданий, анализа цифрового следа, адаптивного сопровождения и поддержки преподавателей разрабатываются изолированно. В связи с этим актуальна разработка универсальных интеллектуальных модулей, интегрируемых в цифровые образовательные платформы и обеспечивающих комплексную автоматизацию образовательного процесса.

Цели и задачи исследования

Целью диссертационной работы является исследование и разработка универсальных интеллектуальных модулей, архитектуры и принципов их интеграции в цифровые образовательные платформы, направленных на обеспечение высокого уровня автоматизации образовательного процесса, снижение рутинной нагрузки преподавателей и повышение эффективности обучения.

Для достижения поставленной цели решаются следующие задачи:

1. Провести анализ современных технологий автоматизации образовательного процесса, включая методы построения архитектуры и разработки цифровой образовательной платформы с отдельными интеллектуальными модулями, в том числе использующими методы искусственного интеллекта, а также разработать методы интеграции различных цифровых образовательных сред;
2. Разработать нейросетевой модуль проверки свободных текстовых ответов обучающихся, основанный на семантической проверке с использованием построения и обработки графов синтаксического разбора предложения;
3. Разработать нейросетевых ассистентов для контекстной помощи обучающимся при решении задач и помощи преподавателям при создании обучающих материалов и проведении занятий, обеспечивающих повышение эффективности образовательного процесса;
4. Разработать модуль поведенческой аналитики и адаптивного обучения, включающий сбор и обработку цифрового следа обучающихся, генерацию синтетических данных на основе математического понятия сигнатур путей и теоремы об оценке погрешности при обращении сигнатуры для построения индивидуальных траекторий обучения с применением методов искусственного интеллекта.

Научная новизна

1. Предложен метод интеграции универсальных интеллектуальных модулей и цифровых образовательных сред в образовательные платформы на основе предложенной архитектуры их взаимодействия, позволяющей расширять

функциональные возможности платформ без модификации их базовых компонентов. Метод обеспечивает повышение уровня автоматизации образовательного процесса и предусматривает применение методов искусственного интеллекта при реализации отдельных модулей.

2. Разработан метод семантической проверки свободных текстовых ответов обучающихся на основе глубокого лингвистического и синтаксического анализа с построением и обработкой графов синтаксического разбора. Метод учитывает синонимы, антонимы, область действия отрицаний и структурные соответствия синтаксических конструкций, что повышает точность автоматизированной проверки ответов на русском языке.

3. Разработан метод контекстно-ограниченного взаимодействия с нейросетевыми ассистентами, основанный на автоматическом формировании контекста из структурированных данных образовательного процесса. Метод предусматривает отбор и структурирование передаваемой языковой модели информации, автоматический выбор режимов работы и контроль допустимости ответа на основе архитектурных и дидактических ограничений, обеспечивая методически релевантную поддержку и соблюдение принципов академической честности.

4. Предложен метод представления цифрового следа обучающихся в виде сигнатур путей, включающий алгоритм обращения сигнатуры и теорему об оценке погрешности аппроксимации непрерывных траекторий дискретными путями. Метод позволяет учитывать порядок, взаимосвязи и динамику показателей цифрового следа, расширяя признаковое пространство и повышая информативность данных для моделей машинного обучения, а также обеспечивает генерацию статистически подобных образовательных траекторий для расширения обучающих выборок.

Теоретическая значимость работы заключается в разработке методов интеграции универсальных интеллектуальных модулей и семантического анализа свободных текстовых ответов на основе графов синтаксического разбора, а также в применении сигнатурных методов для представления и обработки цифрового

следа обучающихся и доказательстве теоремы об оценке погрешности при обращении сигнатуры.

Практическая значимость работы заключается в применении результатов диссертационной работы для создания основных интеллектуальных модулей, интегрированных и апробированных на базе цифровой образовательной платформы Мирера, внедренной в образовательный процесс в 6 организациях.

Объектом исследования являются процессы автоматизации образовательной деятельности, реализуемые посредством программных решений и цифровых технологий.

Предметом исследования являются набор универсальных интеллектуальных модулей и архитектура их интеграции в цифровую образовательную платформу.

Методология исследования включает методы архитектурного и программно-алгоритмического проектирования, машинного обучения, обработки естественного языка и статистического анализа.

Положения, выносимые на защиту

1. Разработанные архитектура взаимодействия и универсальные интеллектуальные модули для цифровой образовательной платформы, использующие методы искусственного интеллекта, обеспечивают автоматизацию образовательного процесса и интеграцию цифровых образовательных сред.

2. Разработанный нейросетевой модуль проверки свободных текстовых ответов обучающихся позволяет проводить семантический анализ с использованием построения и обработки графов синтаксического разбора предложения с учётом синонимов, антонимов, области действия отрицаний и структурных соответствий синтаксических конструкций.

3. Разработанные нейросетевые ассистенты на основе контекстно-ограниченного взаимодействия обеспечивают поддержку обучающихся при решении учебных задач и помогают преподавателям в создании материалов и проведении занятий, обеспечивая повышение эффективности образовательного процесса.

4. Разработанный модуль поведенческой аналитики и адаптивного обучения, использующий метод представления цифрового следа обучающихся в виде сигнатур путей, алгоритм обращения сигнатуры и теорему об оценке погрешности, обеспечивает сбор и обработку цифрового следа обучающихся, учёт порядка, взаимосвязей и динамики его показателей, расширяет и повышает информативность данных для моделей машинного обучения, а также обеспечивает генерацию синтетических образовательных траекторий.

Соответствие специальности

Диссертация соответствует паспорту научной специальности 2.3.8. Информатика и информационные процессы и охватывает следующие области исследования, входящие в эту специальность:

п. 1. «Разработка компьютерных методов и моделей описания, оценки и оптимизации информационных процессов и ресурсов, а также средств анализа и выявления закономерностей на основе обмена информацией пользователями и возможностей используемого программно-аппаратного обеспечения.»

п. 5. «Лингвистическое обеспечение информационных систем и процессов. Методы и средства проектирования словарей данных, словарей индексирования и поиска информации, тезаурусов и иных лексических комплексов. Методы семантического, синтаксического и прагматического анализа текстовой информации для представления в базах данных и организации интерфейсов информационных систем с пользователями.»

п. 16. «Автоматизированные информационные системы, ресурсы и технологии по областям применения (научные, технические, экономические, образовательные, гуманитарные сферы деятельности), форматам обрабатываемой, хранимой информации. Системы принятия групповых решений, системы проектирования объектов и процессов, экспертные системы и др.»

Публикации

По теме диссертации опубликовано 28 научных работ, в том числе 5 работ в изданиях, рекомендуемых ВАК, 2 работы в рецензируемых изданиях, входящих в международные базы данных Web of Science / Scopus, 1 работа является главой в

коллективной монографии. 3 работы выполнены без соавторов. Получено 2 свидетельства о регистрации программы для ЭВМ (Приложение А).

Степень достоверности, апробация и внедрение результатов работы

Основные результаты диссертации были доложены на 14-ти международных научных конференциях, в том числе:

- Информатизация образования и методика электронного обучения: цифровые технологии в образовании, Красноярск, 2020, 2022, 2024, 2025;
- Новые образовательные стратегии в современном информационном пространстве, Санкт-Петербург, 2023;
- Новые образовательные стратегии в открытом цифровом пространстве, Санкт-Петербург, 2024, 2025, 2026;
- СПО: от обучения до разработки, Переславль-Залесский, 2022;
- Физико-математические, естественно-научные и социальные аспекты современного развития науки, техники и общества, Казань, 2024;
- Наука и образование в обеспечении устойчивого развития человеческого потенциала в условиях перехода к цифровой экономике, Пермь, 2024, 2025;
- Сквозные технологии в преподавании физики, математики, информатики, Пермь, 2026;
- Фундаментальные проблемы обучения математике, информатике и информатизации образования, Елец, 2024.

Основные результаты диссертационной работы использовались при выполнении:

- Госзадания по теме № FNEF-2022-0010;
- Госзадания по теме № FNEF-2024-0001 (1023032100070-3-1.2.1);
- Гранта РФФИ 19-29-14057.

Разработанные в рамках диссертационной работы универсальные интеллектуальные модули, интегрированные в цифровую образовательную платформу Мирера, внедрены в образовательный процесс в следующих учреждениях (Приложение Б):

- ФГБОУ ВО «Российская академия народного хозяйства и государственной

службы при Президенте Российской Федерации»;

– ФГБОУ ВО «Финансовый университет при Правительстве Российской Федерации»;

– АНО ДПО «Институт образовательных технологий»;

– ФГАОУ ВО «Казанский (Приволжский) федеральный университет»;

– ФГБОУ ВО «Государственный университет управления»;

– ФГБОУ ВО «Московский педагогический государственный университет».

Структура диссертации

Диссертация состоит из введения, четырёх глав, заключения, списка литературы, включающего 165 наименований, и двух приложений. Объем работы составляет 171 страницу машинописного текста.

ГЛАВА 1. АНАЛИЗ ЭВОЛЮЦИИ И ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ПЛАТФОРМ С РАЗЛИЧНЫМ УРОВНЕМ АВТОМАТИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Современный рынок труда выдвигает требования, при которых качественное базовое образование уже не гарантирует трудоустройство, особенно в прикладных технических областях. Это вызвано ускоряющимся развитием и быстрой сменяемостью технологий в условиях повсеместного распространения информационно-коммуникационных технологий и методов искусственного интеллекта. Такая ситуация предъявляет новые требования, в том числе и к системе образования, которая должна быть нацелена не только на обучение в учебных заведениях, но и способна предложить работодателям технологии оперативного дообучения, переобучения и непрерывного поддержания уровня компетенции сотрудников в активном и актуальном состоянии.

При этом использование цифровых образовательных платформ не подразумевает исключительно дистанционное образование, они могут использоваться для поддержки учебного процесса при классическом очном обучении. Цифровые образовательные платформы с элементами автоматизации – это современные инструменты обучения, которые позволяют ученикам и преподавателям эффективно использовать технологии для повышения качества образования. Развитие подобных платформ имеет важное значение в современном образовании, так как они позволяют учиться в любое время и в любом месте, а также предоставляют доступ к большому количеству учебных материалов.

Академик А.П. Ершов говорил: «Если выяснится, что какого-то учителя можно заменить компьютером, то это нужно сделать немедленно» [28]. Однако, несмотря на распространенное мнение, что системы электронного обучения призваны заменить преподавателей, на сегодняшний день без участия реального педагога невозможно предоставить качественное образование [11; 32]. Преподаватель играет важную роль при использовании цифровых

образовательных платформ, так как он должен выбирать подходящие материалы и курсы, следить за успеваемостью и помогать отстающим студентам, адаптировать учебный процесс под разный уровень обучающихся. Помимо этого, однажды разработанный курс могут впоследствии проходить десятки тысяч обучающихся, что значительно разгружает преподавателя в необходимости постоянно повторять один и тот же материал и позволяет ему лишь вносить в него корректировки. Студенты же имеют возможность решать большое количество практических заданий и тем самым более качественно и равномерно усваивать материал, поскольку решения проверяются автоматически и на их проверку не уходит время преподавателя.

1.1 Эволюция автоматизированных и интеллектуальных обучающих систем

В прошлом существовал широкий спектр автоматизированных систем обучения различного назначения [17; 33]. В развитии обучающих систем можно выделить этапы, каждый из которых характеризуется новыми подходами и технологиями [51].

В работах С.Л. Пресси и Б.Ф. Скиннера были заложены ключевые концепции автоматизации обучения, включающие пошаговое освоение материала, автоматическую проверку заданий и мгновенную обратную связь, которые впоследствии стали базовыми принципами современных цифровых образовательных платформ [125; 73; 162].

С развитием компьютеров появились первые компьютеризированные обучающие системы (программно-аппаратные обучающие системы, автоматизирующие отдельные функции учебного процесса), которые разрабатывались с целью масштабирования и удешевления образования за счёт автоматизации отдельных функций учебного процесса, повышения доступности обучения и снижения потребности в большом количестве квалифицированных преподавателей, исходя из идеи переноса принципов автоматизации производственных процессов на образовательную сферу [158]. Система PLATO [145], разработанная Д. Битцером, позволяла создавать учебные курсы на языке

TUTOR и анализировать ответы студентов с использованием технологии сопоставления с шаблоном [141]. Другой системой того времени была TISCIT [132], которая позволяла студентам выбирать порядок изучения материала.

С развитием методов искусственного интеллекта начали появляться интеллектуальные системы обучения (автоматизированные обучающие системы с поддержкой персонализированного обучения). Одной из первых таких систем была SCHOLAR [53], которая могла давать обратную связь на естественном языке. Данная система использовала диалоговый режим взаимодействия с обучаемым и описывала данные в виде семантической сети. Система анализировала ответы студентов, классифицировала ошибки и адаптировала учебную программу в зависимости от прогресса обучаемого. В этот период также появились первые попытки создания адаптивных систем, которые могли изменять порядок изучения материала в зависимости от успехов студента.

Позднее происходит смена существовавшей парадигмы на новую: «индивидуальный репетитор каждому обучаемому». Перенос систем на персональные компьютеры повысил их интерактивность и удобство. Системы были направлены на формирование процедурных знаний (развитие у обучающегося навыков решения задач на практике) и поддержку решения задач с применением технологий анализа ошибок, подсказок и примеров [51; 144]. Распространение персональных компьютеров способствовало популяризации образовательного ПО, появлению обучающих игр и систем управления обучением (LMS), таких как FirstClass [127].

С распространением Интернета начали появляться онлайн-обучающие системы, которые становятся доступными на персональных компьютерах и мобильных коммуникаторах. В связи с этим появляются новые области для исследования, такие как поддержка совместной работы и удаленного взаимодействия с преподавателем. Первой платформой, объединившей возможности Интернета и обучения, стала WebCT. Она позволяла создавать виртуальные учебные среды – интерактивные пространства, где студенты могут взаимодействовать друг с другом, изучать материалы курса, проходить

тестирования и т.п. В этот период также был разработан стандарт SCORM, который стал основой для создания переносимых учебных курсов. Система E-Tutor, разработанная для обучения немецкому языку, использовала элементы адаптивного обучения и персонализации, анализируя действия обучаемого и подстраивая материалы и задания под его прогресс [88].

Современные интеллектуальные обучающие системы обеспечивают поддержку массового обучения и сбор данных о процессе обучения, а также используют методы анализа больших данных и адаптивной генерации контента, включая прогнозирование трудностей обучаемых, выявление закономерностей в их действиях и автоматическую подстройку учебных материалов под индивидуальные потребности студента [89; 96; 45; 58]. Некоторые современные системы включают мониторинг активности, вовлечённости и успеваемости обучающихся для поддержки адаптивного обучения [67; 75; 81], однако ряд исследователей критикуют такой подход как неэтичный, отмечая, что постоянный контроль может вызывать стресс и негативно влиять на психическое здоровье обучающихся. Одним из векторов развития в онлайн-образовании стали массовые открытые онлайн-курсы (МООС, англ. Massive Open Online Course), которые позволяют обучать тысячи студентов одновременно, однако их эффективность остается предметом дискуссий [21].

1.2 Сравнительный анализ современных цифровых образовательных платформ и интеллектуальных модулей автоматизации образовательного процесса

Современные цифровые образовательные платформы представляют собой комплексные программные системы, обеспечивающие поддержку различных аспектов образовательного процесса, включая управление учебным контентом, контроль знаний, коммуникацию участников и анализ образовательной деятельности. При этом одним из ключевых направлений их развития является внедрение интеллектуальных методов автоматизации, основанных на технологиях искусственного интеллекта. В связи с этим проводить анализ и сравнение

цифровых образовательных платформ необходимо не изолированно, а в совокупности с реализуемыми в них интеллектуальными модулями, обеспечивающими сопровождение и автоматизацию образовательного процесса. Анализ существующих решений позволяет выделить типовой набор модулей, характерных для образовательных платформ:

1. Модули управления обучением (LMS-ядро), включающие средства создания и управления курсами, группами и образовательным контентом;
2. Модули оценки и контроля знаний, оценивание и сопровождение учебной деятельности;
3. Модули коммуникации, обеспечивающие взаимодействие участников образовательного процесса, включая чаты, форумы, видеоконференции и другие средства синхронного и асинхронного общения;
4. Модули автоматической проверки программного кода, обеспечивающие автоматизированную проверку решений задач по программированию;
5. Модули автоматической проверки открытых текстовых ответов, основанные на методах искусственного интеллекта;
6. Модули адаптивных образовательных траекторий, обеспечивающие персонализацию содержания и последовательности обучения;
7. Модули аналитики, предназначенные для сбора, хранения и обработки данных о действиях обучающихся (цифрового следа), а также формирования отчетов и визуализации образовательных результатов;
8. Модули интеллектуальной поддержки, включающие средства генерации учебного контента, а также интеллектуальных ассистентов;
9. Модули интеграции, обеспечивающие взаимодействие платформы с внешними информационными системами, образовательными средами.

Дальнейший сравнительный анализ цифровых образовательных платформ проводится в разрезе наличия и степени развития указанных функциональных модулей, что позволяет выявить не только возможности систем, но и ограничения, препятствующие автоматизации образовательного процесса.

Среди международных платформ дистанционного обучения одной из наиболее распространённых является платформа Moodle [29; 27; 4]. Платформа реализует развитое LMS-ядро, модули контроля знаний и коммуникации, а также обладает высокой расширяемостью за счёт открытого исходного кода и системы плагинов (подключаемых расширений). Вместе с тем аналитические возможности Moodle в базовой конфигурации ограничены преимущественно отчетностью и не обеспечивают глубокого анализа цифрового следа обучающихся. Поддержка адаптивных образовательных траекторий реализуется фрагментарно через систему условий и внешние плагины. В последние годы для Moodle начали появляться внешние AI-модули автоматической проверки открытых ответов и генерации обратной связи, однако они не входят в стандартную архитектуру платформы и требуют дополнительной интеграции. Аналогично автоматическая проверка программного кода обеспечивается преимущественно внешними сервисами. Практическое применение Moodle также выявляет высокую сложность администрирования и необходимость привлечения технических специалистов. Частичным решением обозначенных проблем выступает локализованная версия «Русский Moodle 3К» [30]. Она предоставляет услуги по внедрению системы и техническому сопровождению преподавателей, что снижает необходимость привлечения узкопрофильных специалистов. Однако зависимость от зарубежной разработки сохраняется, поскольку Moodle поддерживается международной командой под руководством австралийского фонда, что делает её уязвимой к санкционным рискам.

Среди российских решений можно выделить системы Modeus, In Study и Эквио, в значительной степени адаптированные к потребностям и специфике работы высших учебных заведений. Платформа Modeus [109] включает развитое LMS-ядро и инструменты управления образовательным процессом, поддерживает индивидуализацию обучения и формирование образовательных траекторий, а также обладает развитой технической поддержкой, что делает её востребованной в крупных вузах. Платформа In Study [93] содержит конструктор курсов и тестов, электронный журнал оценок, встроенный мессенджер и планировщик занятий,

автоматически собирает статистику успеваемости и поддерживает персонализированные траектории обучения. Платформа Эквио [34] ориентирована на корпоративное обучение и имеет гибкие инструменты создания и управления контентом, включая офлайн-доступ. Все перечисленные платформы реализуют базовые модули управления обучением, оценки знаний и коммуникации, однако аналитические возможности ограничиваются агрегированными показателями успеваемости без возможности глубокого анализа цифрового следа и прогнозирования. Поддержка адаптивных траекторий носит преимущественно организационный, а не интеллектуальный характер. Модули автоматизации и интеллектуальной поддержки практически отсутствуют: автоматическая проверка ограничена тестовыми заданиями закрытого типа, средства проверки программного кода и открытых текстовых ответов не реализованы, интеллектуальные ассистенты отсутствуют.

Одной из наиболее распространённых цифровых образовательных платформ в мировом масштабе является Google Classroom [92]. Платформа предоставляет базовое LMS-ядро и обладает тесной интеграцией с сервисами Google, включая Docs, Drive и Gmail, что упрощает организацию учебного процесса, обмен файлами, управление заданиями и коммуникацию между участниками образовательного процесса. В платформе отсутствуют средства адаптивного обучения и механизмы автоматической проверки сложных заданий, интеллектуальные ассистенты ограничены, а интеграционные возможности замкнуты на экосистему Google, что снижает её применимость в условиях высшего образования.

Среди отечественных платформ массовых открытых онлайн-курсов следует выделить Универсариум [31]. Платформа предоставляет бесплатный доступ к большому количеству курсов по различным дисциплинам, разработанных при участии ведущих российских образовательных организаций. Каждый курс включает видеолекции, домашние задания и тестовые материалы. Платформа реализует базовые компоненты LMS-ядра, включая управление курсами и образовательным контентом, средства оценки и контроля знаний, а также базовые

инструменты коммуникации в формате форумов и обсуждений. Вместе с тем аналитические возможности платформы ограничиваются базовыми показателями успеваемости, интеллектуальные модули отсутствуют, а интеграционные возможности не позволяют эффективно встраивать курсы в образовательный процесс вузов без дополнительной технической доработки.

Для корпоративного обучения и дополнительного образования востребованы такие платформы, как Eduson и Skillspace. Eduson [19] фокусируется на развитии soft skills, сочетая LMS с библиотекой курсов и симуляторами. Skillspace [143] ориентирована на создание и сопровождение онлайн-курсов, обеспечивая базовые LMS-функции, тестирование, проведение вебинаров, интеграции с внешними сервисами и инструменты совместной работы преподавателей и обучающихся, однако многие функции доступны только на платной основе.

В российских вузах также находит применение международная платформа Blackboard [2], адаптированная для дистанционного обучения. Платформа предоставляет развитую инфраструктуру дистанционного обучения, включая инструменты видеоконференций, календарное планирование и отслеживание активности обучающихся. Blackboard создаёт комплексную виртуальную учебную среду, что делает её привлекательной для крупных университетов. Ключевым преимуществом является наличие встроенного интеллектуального ассистента, позволяющего автоматизировать создание учебных материалов.

На уровне школьного образования в России широко применяется платформа Якласс [3; 6], поддерживаемая Министерством просвещения РФ. Она предлагает стандартизированные материалы в соответствии с учебной программой, что обеспечивает преемственность в обучении. Платформа характеризуется глубокой интеграцией технологий искусственного интеллекта: используются методы компьютерного зрения для анализа рукописного текста, NLP-модели для оценки открытых ответов и формирования подсказок, а также алгоритмы машинного обучения, обеспечивающие адаптивное изменение сложности заданий и построение индивидуальных образовательных траекторий.

Существенным представителем российского сегмента цифровых

образовательных платформ для средней школы является СберКласс. Платформа представляет собой комплексное решение для школьного образования, включающее модули управления обучением, контроля знаний, коммуникации и мониторинга образовательных результатов. Система поддерживает формирование персонализированных траекторий обучения с учётом интересов и уровня подготовки учащихся. Кроме того, в платформу интегрирован цифровой AI-ассистент, напоминающий о сроках выполнения заданий, помогающий планировать учебную деятельность и поддерживающий мотивацию.

В современном образовательном процессе, особенно в сфере информационных технологий и программирования, широкое распространение получили специализированные платформы, среди которых можно выделить ejudge, Яндекс.Контест и Codeforces. Данные системы предоставляют инструменты для организации соревнований, автоматизированной проверки решений и сопровождения учебных курсов. Система ejudge [25], ориентированная на проведение олимпиад и курсов по программированию, обладает широкими возможностями автоматической проверки программных решений, однако отличается высокой сложностью настройки и администрирования, что требует от преподавателя достаточно глубоких технических компетенций. Дополнительным ограничением являются недостаточно развитые средства управления курсами и разграничения прав доступа между преподавателями. Платформа Яндекс.Контест [26] предназначена для проведения онлайн-соревнований и проверки задач по программированию. Система поддерживает более двадцати языков программирования и применяется как в образовательных организациях, так и при проведении крупных соревнований. Платформа Codeforces [24] широко используется в сообществе соревнований по алгоритмическому программированию и обеспечивает развитый модуль оценки и контроля знаний с рейтинговой системой. Ключевыми преимуществами являются масштабируемость и развитое открытое API для интеграции. Вместе с тем модули управления обучением не обеспечивают полноценного структурирования курсов и индивидуальных траекторий, аналитические возможности платформы базовые, а

интеллектуальные модули отсутствуют. Платформа не предназначена для использования в качестве основной LMS в образовательных учреждениях.

В современном онлайн-образовании платформы дистанционного обучения, ориентированные на массовые онлайн-курсы и самостоятельное обучение, такие как Stepik, Открытое образование и Coursera, обеспечивают доступ к образовательным материалам и развитие профессиональных и академических навыков. Однако ограничения в области управления контентом, взаимодействия участников образовательного процесса, аналитики и технической интеграции могут снижать эффективность обучения. Stepik [146] ориентирован на преподавание технических дисциплин, включая программирование и анализ данных, поддерживает автоматическую проверку программных и SQL-заданий, а также создание мультимедийных курсов с видео, тестами и интерактивными заданиями. Вместе с тем аналитические возможности платформы ограничены базовой статистикой, а интеллектуальные модули отсутствуют. Платформа Открытое образование [20], созданная ведущими российскими университетами, реализует базовые функции управления обучением и оценки знаний. Ключевым преимуществом является высокое качество контента, разработанного авторитетными вузами, и соответствие образовательным стандартам. Однако коммуникационные возможности ограничены форумами, аналитические и интеллектуальные модули отсутствуют, интеграция в образовательный процесс вузов затруднена, а встроенные средства автоматической проверки заданий по программированию не предусмотрены. Coursera [63] является глобальной платформой онлайн-обучения, предлагающей доступ к курсам и программам от ведущих университетов и компаний. Она предоставляет разнообразные форматы обучения, включая видео-лекции, задания и проекты. В рамках Coursera for Campus реализованы функции интеграции курсов в образовательный процесс, управления группами и мониторинга прогресса обучающихся. Платформа включает интеллектуальные модули автоматизации, такие как AI-ассистент Coursera Coach и AI-assisted grading для автоматической проверки развёрнутых ответов, частично реализована проверка задач по программированию. При этом аналитика

ограничивается отслеживанием активности и успеваемости без полноценного анализа цифрового следа и предиктивных моделей. Существенным ограничением для российских образовательных организаций остаётся зависимость от зарубежной инфраструктуры. Кроме того, все подобные платформы ориентированы преимущественно на самостоятельное обучение, тогда как отсутствие активного участия преподавателя снижает долю студентов, успешно завершающих курсы.

Таким образом, проведённый анализ современных цифровых образовательных платформ показывает, что, несмотря на широкое разнообразие реализуемых функциональных модулей, большинство существующих решений ограничивается базовыми возможностями управления обучением, контроля знаний и коммуникации. При этом аналитические инструменты в основном носят описательный характер, а модули автоматизации и интеллектуальной поддержки либо отсутствуют, либо реализованы фрагментарно и не обеспечивают комплексной интеграции методов искусственного интеллекта в образовательный процесс. В связи с этим особую актуальность приобретает дальнейший анализ цифровых образовательных решений, в которых технологии искусственного интеллекта играют ключевую роль.

Использование технологий искусственного интеллекта позволяет автоматизировать значительную часть рутинных задач преподавателя, включая проверку заданий, генерацию учебных материалов и предоставление первичной обратной связи обучающимся. Это позволяет преподавателю сфокусироваться на задачах, требующих профессиональной экспертизы: актуализации и развитии учебных программ, индивидуальной поддержке обучающихся и общем повышении качества образовательного процесса. Обучающимся применение технологий искусственного интеллекта позволяет получать мгновенную обратную связь и поддержку, а также обеспечивает реализацию адаптивных траекторий и большего количества различных заданий, что значительно увеличивает эффективность обучения. Помимо этого, интеллектуальные системы используются для автоматизации проверки отдельных типов задач, снижения субъективности оценивания, оперативного выявления сложных для освоения элементов курса, а

также предоставления преподавателю предиктивной аналитики по курсу [9; 10]. Следовательно, технологии искусственного интеллекта целесообразно рассматривать не как средство замещения преподавателя, а как инструмент повышения качества образования с помощью развития интерактивности, адаптивности и персонализации образовательного процесса.

Интеллектуальная обучающая система SCHOLAR была первой системой, которая обеспечивала индивидуальные траектории обучения и поддерживала диалог со студентами на естественном языке по теме занятий [53]. Система стала пионерской в области использования ИИ в образовании и заложила основы для развития современных интеллектуальных обучающих систем. Современными примерами диалоговых обучающих систем являются AutoTutor [120], симулирующая диалог между преподавателем и студентом во время выполнения заданий, WatsonTutor [150], разработанная Pearson и IBM для вузов и адаптирующая диалог в зависимости от прогресса. Несмотря на наличие развитых механизмов обработки естественного языка и адаптации образовательного взаимодействия, данные системы не обеспечивают управление образовательным процессом и не интегрированы в состав комплексных образовательных платформ.

Среди систем, созданных для увеличения скорости и объективности оценивания при проверке больших письменных заданий, можно выделить систему Revision Assistant компании Turnitin, которая автоматически проверяет эссе, основываясь на анализе комментариев, написанных экспертами ранее, а также систему OpenEssayist, разработанную Открытым университетом и Оксфордским университетом, которая дает студентам развернутую обратную связь по письменному заданию. В Институте русского языка им. Пушкина разработали систему автоматической проверки заданий по русскому языку, которая охватывает как простые правила орфографии, так и более сложные задания на выбор подходящих фраз в контекст предложения. Однако все эти системы позволяют проверять только заранее известный набор заданий, что ограничивает возможность использования данных систем преподавателями и разработчиками курсов в рамках своих собственных, новых заданий без привлечения специалистов по машинному

обучению или лингвистической формализации.

В вопросах применения технологий адаптивного обучения можно выделить онлайн-систему 01Математика [35], созданную для персонализированного обучения школьников, которая уже сейчас становится одной из базовых обучающих систем по математике в школах. Система реализует развитые модули оценки знаний и адаптивного управления. Модуль автоматизации и интеллектуальной поддержки включает четыре уровня адаптации, предсказание времени выполнения задач и персонализацию на основе анализа успешных траекторий других пользователей. Модуль аналитики обеспечивает сбор статистики, учёт кривой забывания и формирование рекомендаций. Среди систем, которые также используются для мониторинга результатов студентов, но с акцентом на раннее прогнозирование итоговых результатов студентов с целью своевременного воздействия на них, можно выделить системы CourseSignals [47], Систему раннего прогнозирования [114], а также Систему прогнозирования академической успеваемости «Пифия» [91].

Проведённый сравнительный анализ показывает, что большинство современных образовательных платформ реализуют базовые модули управления обучением, контроля знаний и коммуникации. В то же время аналитические возможности в значительной части систем ограничиваются формированием отчетности и не обеспечивают глубокого анализа цифрового следа обучающихся. При этом специализированные системы демонстрируют высокий уровень автоматизации отдельных задач (например, проверки программных решений или текстов), однако не интегрированы в образовательные платформы как единый функциональный контур. Итоговые результаты сравнения рассмотренных образовательных платформ в разрезе наличия и степени развития приведенных функциональных модулей представлены в таблице, Таблица 1.

Таблица 1 – Результаты обзора образовательных платформ

Платформа	LMS-ядро	Оценка и контроль	Общение	Проверка кода	Проверка текстов	Адаптивные траектории	Аналитика	AI-модули	Интеграции	Разработано в РФ
Moodle	+	+	+	+-	+-	+-	+-	+-	+	-
Modeus; In Study; Эквио	+	+	+	-	-	+	+-	-	+-	+
Google Classroom	+	+	+	-	-	-	+-	+-	+-	-
Универсариум	+	+	+-	-	-	+-	+-	-	-	+
Eduson; Skillspace	+	+	+-	-	-	+	+	-	+	+
Blackboard	+	+	+	-	+-	+-	+	+	+	-
Якласс	+-	+	+-	-	+	+	+	+	+-	+
СберКласс	+	+	+-	-	-	+	+-	+	+-	+
ejudge	+-	+	-	+	-	-	-	-	+-	+
Яндекс.Контест	+-	+	+-	+	-	-	+-	-	+-	+
Codeforces	-	+	+	+	-	-	+-	-	+-	+
Stepik	+	+	+-	+	-	+-	+-	-	+	+
Открытое образование	+-	+	+-	-	-	-	-	-	+-	+
Coursera	+	+	+	+-	+-	+	+-	+	+	-
AutoTutor; WatsonTutor	-	+-	+	-	+	+	+-	+	+-	-
Revision Assistant; OpenEssayist	-	+	+-	-	+	+-	+-	+	+-	-
01 Математика; Пифия	-	+	-	-	-	+	+	+	+-	+

1.3 Ограничения существующих решений и обоснование необходимости разработки новой отечественной цифровой образовательной платформы с универсальными интеллектуальными модулями

В современной организации высшего образования, цифровых образовательных платформах и их использовании можно выделить следующие проблемы. На преподавателях лежит высокая рутинная нагрузка, до 80% времени педагога уходит на проверку заданий и формирование отчетности. В связи с этим у него остается мало времени на индивидуальную работу со студентами и совершенствование педагогических методик. Из-за этого падает вовлеченность студентов в образовательный процесс и их успеваемость. Студенты теряют интерес из-за недостатка практических заданий и персонализированной обратной связи, отстающие студенты начинают списывать и не завершают успешно курс.

Использование современных цифровых образовательных платформ, которое могло бы решить эти проблемы, в реальной практике часто приводит к обратному эффекту. Создание и обновление курсов требует значительных временных затрат и технических навыков от преподавателя, время, потраченное на цифровизацию своего курса, становится больше, чем эффект от использования образовательной платформы. Помимо этого, у преподавателей возникает страх потери интеллектуальной собственности, онлайн-курсы пытаются заместить преподавателя и могут проводиться без него, однако отсутствие преподавателя снижает итоговое качество образования студентов.

Вместе с этим, существующие платформы не обеспечивают полноценной автоматизации проверки заданий и других рутинных процессов в образовании, в связи с чем необходимость их использования для реального преподавателя остается под вопросом. Несмотря на большое количество собираемых данных цифрового следа студентов, у существующих образовательных платформ недостаточно инструментов для построения глубокой, многоуровневой поведенческой аналитики и персонализации образовательных траекторий.

Для преподавателей есть ряд ограничений по использованию технологий

искусственного интеллекта, связанных с тем, что практически отсутствуют готовые образовательные решения, которые можно было бы использовать «из коробки», они требуют от преподавателя технических навыков по работе с моделями машинного обучения и не интегрированы в образовательные платформы [8; 10], из-за чего нужно искать различные модели, адаптировать их под образовательные задачи, после чего остается рутинная нагрузка по переносу полученных результатов в образовательную платформу. Решением могло бы быть создание интеллектуальных модулей образовательной платформы, в которой технологии искусственного интеллекта встроены на уровне базовой архитектуры, чтобы для преподавателя их использование ограничивалось бы простым интерфейсным взаимодействием.

В связи с отказом со стороны зарубежных разработчиков поддержки установленных в России иностранных образовательных платформ, при их использовании есть риски потери разработанных курсов и методик интеллектуальной собственности, а также блокировки используемого в учебном процессе программного обеспечения зарубежной разработки, что может парализовать работу вузов. В современных условиях усиления внешнеполитической напряженности и санкционных ограничений перед российскими вузами остро встает вопрос выбора цифровых образовательных платформ, соответствующих требованиям информационной безопасности [14].

Таким образом, является актуальным вопрос разработки отечественной цифровой образовательной платформы, обеспечивающей технологический суверенитет, а также универсальных интеллектуальных модулей, которые, в отличие от используемых сегодня в России импортных систем, будут обладать полным функционалом для проведения интегральной цифровой трансформации образовательного процесса.

Выводы по главе 1

В первой главе проведён анализ эволюции цифровых образовательных платформ, их архитектурных особенностей и функциональных возможностей с

точки зрения уровня автоматизации образовательного процесса. Рассмотрение данной проблематики позволило выявить ключевые тенденции развития обучающих систем, а также определить ограничения существующих решений, препятствующие эффективной цифровой трансформации образования.

В ходе исследования установлено, что развитие цифровых образовательных платформ сопровождалось постепенным переходом от автоматизации отдельных задач, таких как проверка заданий и масштабирование обучения, к построению интегрированных систем поддержки образовательного процесса. Показано, что современные платформы ориентированы на комплексную поддержку образовательного процесса, включая интерактивное взаимодействие, сбор и анализ цифрового следа обучающихся, а также формирование адаптивных образовательных траекторий.

Проведённый функционально-модульный анализ позволил формализовать структуру цифровых образовательных платформ как совокупности взаимосвязанных компонентов, обеспечивающих управление обучением, контроль знаний, коммуникацию, аналитику, интеграцию и интеллектуальную поддержку. Установлено, что уровень развития и согласованности данных модулей напрямую определяет степень автоматизации образовательного процесса и эффективность использования платформ в образовательной практике.

Сравнительный анализ современных решений показал, что большинство из них реализует преимущественно базовые функции и не обеспечивает комплексной автоматизации образовательной деятельности. Выявлено, что аналитические возможности таких систем, как правило, ограничиваются формированием отчётности, а инструменты интеллектуальной поддержки либо отсутствуют, либо представлены фрагментарно и не интегрированы в единую архитектуру платформы. Анализ специализированных образовательных решений также показал, что при высокой эффективности автоматизации отдельных задач они не обеспечивают целостной поддержки образовательного процесса и приводят к фрагментации используемых инструментов. Отдельное внимание уделено исследованию применения технологий искусственного интеллекта в образовании.

Установлено, что существующие интеллектуальные решения обладают высоким потенциалом, однако требуют значительных технических компетенций и редко представлены в виде встроенных инструментов, доступных преподавателю непосредственно в образовательной платформе.

Анализ реальной образовательной практики показал, что ключевой проблемой остаётся высокая рутинная нагрузка преподавателей, связанная с проверкой заданий, организацией учебного процесса и ведением отчетности. При этом существующие цифровые платформы не обеспечивают её существенного снижения, а в ряде случаев приводят к увеличению трудозатрат за счёт сложности внедрения и эксплуатации. Дополнительно выявлены риски, связанные с использованием зарубежных образовательных решений, включая зависимость от внешней инфраструктуры и потенциальные угрозы информационной безопасности.

В результате проведённого анализа установлен системный разрыв между возможностями современных технологий, в том числе методов искусственного интеллекта, и уровнем их интеграции в цифровые образовательные платформы. Обоснована необходимость перехода к архитектурно целостным решениям, основанным на использовании универсальных интеллектуальных модулей, интегрированных в единую цифровую образовательную платформу и обеспечивающих автоматизацию ключевых процессов образовательной деятельности.

ГЛАВА 2. РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ МОДУЛЕЙ ДЛЯ АВТОМАТИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА, В ТОМ ЧИСЛЕ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

2.1 Модуль учёта посещаемости студентов в условиях очного, дистанционного и гибридного обучения

Распространение смешанных форматов обучения требует точных и масштабируемых систем учёта посещаемости, вовлеченности и активности обучающихся, обеспечивающих прозрачность образовательного процесса. Классические методы контроля посещаемости утрачивают эффективность на фоне новых требований к скорости и достоверности, а также интеграции данных в цифровые образовательные экосистемы, необходимой для дальнейшего построения учебной аналитики, отслеживания динамики образовательного процесса и аналитического сопровождения образовательных траекторий. Традиционные подходы, как правило, предполагают ручной учёт присутствия студентов преподавателем при проведении очных занятий или фиксацию факта входа обучающегося на страницу занятия при дистанционном обучении. При очном обучении такой процесс либо требует значительных временных затрат, либо приводит к появлению большого числа ошибочно отмеченных отсутствующих студентов. В дистанционном формате фиксация исключительно факта входа создаёт возможность формального «пролистывания» занятий без фактического ознакомления с учебными материалами. Кроме того, традиционные методы недостаточно адаптированы к современным гибридным форматам организации обучения.

2.1.1 Анализ существующих подходов к автоматизированному учёту посещаемости

Существующие подходы к автоматизированному учёту посещаемости студентов обладают рядом достоинств и ограничений, влияющих на их практическое применение. Одним из наиболее распространённых решений

являются системы, основанные на использовании RFID-меток [7], а также других бесконтактных технологий идентификации, включая NFC [118]. Подобные системы обеспечивают фиксацию фактов входа и выхода студентов и сотрудников посредством считывания идентификаторов при прохождении контрольных точек. Однако их возможности ограничены: такие системы не позволяют достоверно определить присутствие студента на конкретной паре, поскольку требуют установки оборудования (например, турникетов) в каждой учебной аудитории, что сопряжено с высокими финансовыми затратами на внедрение и техническое обслуживание. Кроме того, данные технологии не обеспечивают оперативный и удобный доступ преподавателей к информации о посещаемости, зачастую плохо интегрируются с цифровыми образовательными платформами, используемыми в учебном процессе, а также вызывают опасения относительно приватности персональных данных [41].

Биометрические системы, в первую очередь основанные на распознавании лиц, демонстрируют высокую потенциальную точность при идентификации личности [13; 140]. Тем не менее, на практике их эффективность существенно снижается в реальных условиях использования, например, плохого освещения, перекрытия лиц, поворота головы или использования масок и аксессуаров, закрывающих части лица. Дополнительно требуется собрать обширную базу изображений каждого студента в различных ракурсах, что вызывает правовые риски, связанные с обработкой персональных данных. Даже при соблюдении всех условий ошибки распознавания остаются вероятными: система может ошибочно зафиксировать присутствие отсутствующего студента или не засчитать посещение присутствующего, при этом быстро обнаружить и скорректировать такие ошибки практически невозможно.

Некоторые решения основаны на использовании мобильных устройств студентов. Такие подходы применяют сигналы GPS, Wi-Fi или Bluetooth для определения местоположения обучающегося [68]. Однако в условиях плотной городской застройки точность геолокации сильно снижается. Кроме того, надёжность метода зависит от добросовестности студентов, например, включения

соответствующего приложения, что не всегда может быть гарантировано.

Помимо технологий фиксации, существуют и программные комплексы, реализующие ручной или полуавтоматический учёт посещаемости. Среди них можно выделить такие системы, как «АРАТО», модуль «Альма Матер» и «1С: Деканат». Комплекс «АРАТО» предоставляет функции электронного журнала и отчетности, однако процесс фиксации посещаемости полностью возлагается на преподавателя, который вручную отмечает студентов, что снижает эффективность автоматизации. Модуль «Альма Матер» позволяет фиксировать сведения о пропущенных занятиях, формируя отчеты на основе сведений, собираемых еженедельно по каждой группе. Система «1С: Деканат», интегрированная с другими модулями 1С, ориентирована в большей степени на административный персонал и не предусматривает активного взаимодействия со стороны преподавателей в процессе проведения занятий.

Общий недостаток всех рассмотренных систем заключается в том, что они либо требуют прямого взаимодействия преподавателя с каждым студентом во время занятия, либо не обеспечивают достаточной точности и надежности в условиях массового применения. Кроме того, их функциональность в дистанционном или гибридном формате обучения остается ограниченной, перекладывая ответственность за организацию учета на преподавателя.

2.1.2 Алгоритмы учёта посещаемости при очных форматах обучения

Разработка модуля учета посещаемости студентов при очной форме обучения требует поиска сбалансированного решения, способного одновременно обеспечить достоверность фиксации присутствия, устойчивость к возможным злоупотреблениям и пригодность к применению в условиях массового обучения. В процессе проектирования и последующего внедрения было реализовано два механизма подтверждения во время очных занятий: по геолокации и с помощью QR-кодов.

Геолокационный метод основан на применении Geolocation API – стандартного инструмента, встроенного в современные браузеры [78]. При начале

первого занятия преподаватель, находясь в аудитории, инициирует запуск соответствующего учебного контеста и отправляет на сервер координаты своего текущего местоположения – широту, долготу и допустимую погрешность в метрах. Эта информация закрепляется за конкретной учебной группой, что исключает необходимость повторной настройки на каждом занятии, а также при копировании занятий между курсами для групп разных университетов. В начале занятия студенты также осуществляют отправку своих геоданных, используя встроенные функции геопозиционирования в браузере на своих мобильных устройствах. Система, сверяя полученные координаты с исходными параметрами контеста, в автоматическом режиме определяет, находится ли пользователь в пределах допустимой зоны, и, в случае положительного результата, засчитывает посещение. Допустимая зона определяется, как погрешность, указанная преподавателем, умноженная на два, для срабатывания в наихудших случаях ошибки определения геопозиции. Такой подход, несмотря на свою концептуальную простоту, продемонстрировал высокую устойчивость в реальных ситуациях и позволил значительно разгрузить преподавателя в вопросах учета посещаемости.

Однако внутри университетских корпусов геолокация часто плохо работает из-за плотной городской застройки, толщины стен и других архитектурных особенностей зданий, что может приводить к существенным искажениям полученных координат геопозиции. В последние годы проблемам стала еще более актуальной в связи с применением средств радиоэлектронной борьбы. Помимо этого, некоторые студенты могут использовать средства подмены геопозиции с помощью сервисов VPN, отправлять свою геопозицию из ближайшей окрестности аудитории, фактически не находясь в ней, что особенно актуально при нахождении общежития рядом с университетским корпусом, а также передавать логин и пароль от своего аккаунта своим одноклассникам для отправки геопозиции вместо них из аудитории. Все эти проблемы вынуждали преподавателей проводить дополнительные переключки и отмечать большой процент студентов вручную, что значительно снижало эффективность применения модуля.

В рамках работы также был предложен и реализован альтернативный подход,

устраняющий часть перечисленных недостатков и основанный на использовании динамических QR-кодов, что позволяет обеспечить быструю и защищенную верификацию очного присутствия. Такой механизм предполагает проведение проверки непосредственно в аудитории и исключает возможность удалённого подтверждения участия в занятии. Разработанный модуль динамически генерирует QR-коды с периодичностью 2 секунды, при этом срок действия каждого кода не превышает 4 секунд. Подобная временная схема обеспечивает строгую привязку к моменту генерации и снижает вероятность повторного использования либо передачи кода отсутствующим студентам, при этом модуль сохраняет работоспособность даже при нестабильном мобильном соединении. Каждый QR-код содержит уникальный токен в формате JSON Web Token (JWT), включающий идентификатор занятия и ограниченный интервал действия, что позволяет гарантированно исключить повторное использование кода и обеспечить корректную идентификацию факта присутствия.

Использование QR-кодов для учёта посещаемости обеспечивает стабильную работу для студентов, действительно присутствующих в аудитории, в отличие от геопозиционного подхода, который мог некорректно функционировать на отдельных устройствах из-за проблем с определением местоположения. С другой стороны, данный метод полностью не устраняет возможность передачи учётных данных присутствующим одноклассникам, а также потенциальную возможность передачи QR-кода отсутствующим студентам по видеосвязи в режиме реального времени. Несмотря на то, что подобный сценарий затруднителен для реализации в условиях реального занятия, технически он остаётся возможным.

2.1.3 Алгоритмы учёта посещаемости при дистанционных форматах обучения

При дистанционном формате обучения в разработанном модуле реализована многоуровневая система учета посещаемости, адаптированная под различные формы учебной активности, включая участие в видеоконференциях, просмотр их записей и взаимодействие с прикрепленными материалами.

Для всех видов обязательных и дополнительных активностей при

дистанционном формате обучения применяется градационная система оценивания, основанная на пропорциональном расчёте. Итоговый балл W в таких случаях определяется в зависимости от фактического времени, проведённого студентом за соответствующим видом активности t_{fact} , по отношению к нормативной продолжительности T_{max} , установленной для конкретного занятия, при этом начисляемые баллы не могут превышать максимально допустимое значение W_{max} , даже если фактическое время участия превышает заданную границу. Итоговый балл W рассчитывается по следующему правилу, формула (1):

$$W = W_{max} \cdot \min\left(1, \frac{t_{fact}}{T_{max}}\right). \quad (1)$$

Центральным элементом системы является модуль видеоконференцсвязи, реализованный на базе Open Source платформы BigBlueButton, обеспечивающий не только проведение онлайн-занятий, но и детализированный мониторинг активности студентов во время их проведения. Для каждого участника автоматически фиксируются ключевые параметры взаимодействия: общее время нахождения в конференции, которое является необходимым условием для засчитывания и выставления баллов при этом типе посещаемости, а также продолжительность включения камеры, количество сообщений, отправленных в чат, и временные интервалы устных ответов преподавателю, все из которых предоставляют возможность получения дополнительных баллов. Помимо этого, собирается время просмотра записей конференций и прикрепленных материалов, например, видео или презентаций, с аналогичной системой градационного начисления баллов.

2.1.4 Интегрированная архитектура учёта гибридной посещаемости

Разработка архитектуры гибридного учёта посещаемости позволила устранить ряд ранее выявленных ограничений обеспечить поддержку гибридного формата обучения. Применение одновременно верификации посещения по геопозиции и по QR-коду при проведении очных занятий значительно повышает устойчивость и удобство системы: геопозиционный подход можно применять для

обработки основного потока обучающихся, после чего точно использовать QR-коды при низкой точности GPS-сигнала. Гибридная архитектура учёта посещаемости также решает задачу фиксации участия студентов при проведении занятий в смешанном формате, когда часть обучающихся присутствует очно, часть подключается дистанционно, а остальные просматривают запись занятия.

Для обеспечения возможности реализации всех этих сценариев была разработана многоуровневая архитектура гибридного учёта посещаемости, обеспечивающая поддержку различных типов верификации участия и гибкость в их конфигурации. Основу системы составляет механизм анализа событий, в котором для каждого вида учебной активности используются отдельные, настраиваемые параметры оценки. Для каждого способа подтверждения посещения преподаватель может независимо задать два показателя: регистрацию в журнале посещаемости, или «галочку посещаемости», которая потом учитывается для итого расчёта процента посещенных студентом занятий, а также количество баллов за данный тип посещаемости, учитываемых в итоговой оценке по курсу.

Таким образом, в рамках данной архитектуры реализована возможность дифференцированной оценки различных форм участия. Так, студенты, присутствующие на занятии очно, могут получать большее количество баллов по сравнению с теми, кто подключается удалённо или ограничивается просмотром записи. При этом факт просмотра записи может фиксироваться в системе, но может быть исключён из расчёта общего процента посещённых занятий, если студент не участвовал в активности во время занятия. Это позволяет учитывать освоение учебного материала в целом, без искажения итоговых показателей посещаемости. На рисунке, Рисунок 1, представлен пример настройки посещаемости гибридного занятия, где включены две альтернативные возможности очного засчитывания посещаемости, за которые ставятся максимальные доступные баллы и галочка учета в журнале посещаемости, а также вариант подключения через конференцию, за которую студент не получит галочку посещаемости, но получит половину доступных баллов и дополнительные баллы при включении камеры по время проведения конференции.

Посещаемость

Баллы Время активности ⌚

Тип посещаемости

► [Геопозиция](#)

▼ [QR-код](#)

☒ Ставить галочку посещаемости

Баллы ☒ по умолчанию

▼ [Видеоконференция](#)

☐ Ставить галочку посещаемости

Баллы ☐ по умолчанию

Дополнительные баллы

☒ Ставить баллы при включенной камере

Баллы Время с включенной камерой (мин.)

☐ Ставить баллы за устный ответ

☐ Ставить баллы за ответ в чате

Рисунок 1 – Пример настроек гибридной посещаемости

Алгоритм начисления баллов реализует принцип приоритизации: при наличии у студента нескольких форм подтверждённого участия в рамках одного учебного события система автоматически фиксирует «галочку» в журнале посещения при наличии хотя бы одного засчитанного и допустимого для этого типа посещаемости. Одновременно из всех полученных баллов за различные типы посещаемости выбирается наибольшее. Дополнительно к этому значению могут быть прибавлены баллы за проявленную активность в ходе занятия. Такие баллы не влияют на статус посещения, однако включаются в итоговую сумму баллов, учитываемую при подсчете баллов за курс. И студентам, и преподавателю подробно отображается отчет о засчитанных типах посещаемости и полученном за них количестве баллов. На рисунке, Рисунок 2, представлен пример отчёта преподавателя, в котором каждый тип учебной активности сопровождается графическим индикатором текущего статуса.

Присутствие на занятии ☒

Первое посещение

13-05-2025 19:38

Комментарий

Баллы 1.18/1

✓ Геопозиция 1/1

Показать на карте

✗ QR-код 0/1

✓ Видеоконференция 0.44/0.5

○ Запись конференции 0/0

Дополнительные баллы

Ответ в чате 0.175/0.25

Баллы от преподавателя 0

Рисунок 2 – Пример отображения отчета посещаемости у преподавателя

Символ в виде галочки обозначает зафиксированную посещаемость, крестик указывает на отсутствие отметки при наличии возможности её получения по данному типу активности, а кружок сигнализирует о недоступности регистрации посещения для соответствующего способа участия. В отчёте также отображаются значения баллов, полученных студентом по каждому из типов посещаемости, а также итоговое количество баллов. Последнее формируется как сумма наибольшего значения среди баллов, начисленных за различные формы участия, и дополнительных баллов, выставленных автоматически за активность во время занятия или вручную преподавателем.

Предложенная архитектура гибридного учёта посещаемости может быть формализована следующим образом, Рисунок 3. Пусть для каждого типа активности заданы параметры, определяющие его вклад в итоговый результат студента. Тогда алгоритм начисления баллов и фиксации посещаемости описывается следующим образом. Пусть для каждого типа учебной активности T_i преподавателем заданы два параметра: $check_i \in \{0,1\}$ – признак, учитывается ли

данный тип при фиксации посещаемости в журнале («галочка»), и $score_i \in \mathbb{R}_{\geq 0}$ – количество баллов, начисляемых за данный тип участия. Множество подтверждённых форм участия студента в рамках одного занятия обозначим как $U \subseteq \{T_1, \dots, T_n\}$. Алгоритм начисления баллов и фиксации посещаемости реализует правила, при которых фиксация посещаемости в журнале определяется наличием хотя бы одного типа участия с разрешённой «галочкой», формула (2):

$$attended = \bigvee_{i \in U} check_i, \quad (2)$$

где $\vee$ обозначает логическое «ИЛИ». Итоговое количество баллов за занятие складывается из максимального балла среди всех подтверждённых типов участия и дополнительных баллов за активность, формула (3):

$$score_{total} = \max_{i \in U} score_i + score_{active}, \quad (3)$$

где $score_{active} \in \mathbb{R}_{\geq 0}$ – сумма баллов за проявленную активность (автоматически или вручную преподавателем).

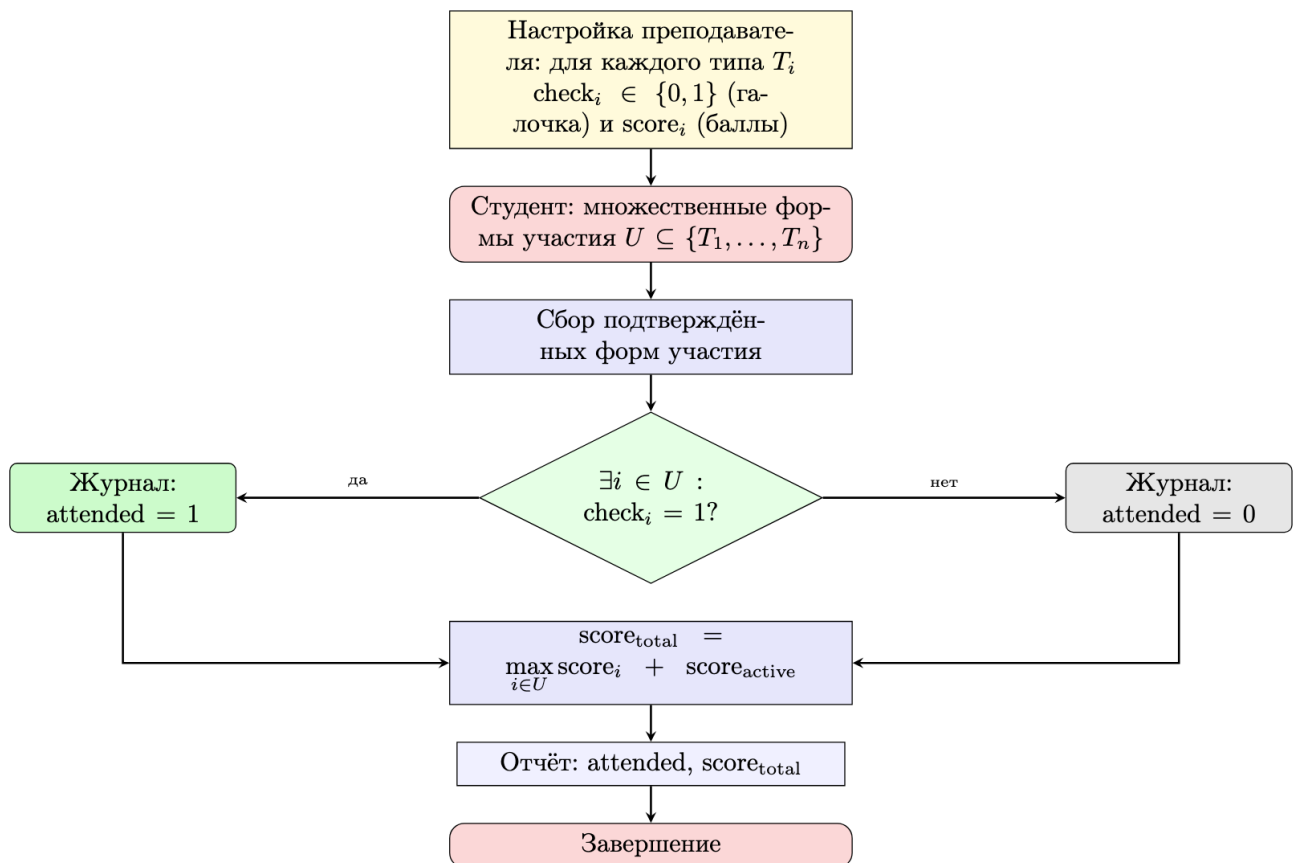


Рисунок 3 – Архитектура гибридного учёта посещаемости

2.1.5 Контрольные метрики и верификация корректности учёта посещаемости по фото

Архитектура автоматизированного учёта гибридной посещаемости, предусматривающая открытый отчёт по всем параметрам фиксации активности, гарантирует, что фактическое присутствие студента на занятии не останется неотмеченным в журнале. Дополнительную мотивацию для студентов обеспечивает возможность ограничения доступа к решению заданий до момента подтверждения посещения. Вместе с тем, такая мера повышает риск попыток со стороны студентов, отсутствующих на занятии, получить несанкционированный доступ с целью фиктивной регистрации посещаемости и получения соответствующих баллов. Таким образом, при условии гарантированной фиксации присутствия каждого студента в журнале посещаемости, подходящей контрольной метрикой может быть пересчет количества людей в аудитории, поскольку исключены случаи, при которых студент присутствует, но не зарегистрирован в системе, а вместо него, напротив, другой студент числится в журнале при фактическом отсутствии на занятии.

При небольшой численности студентов пересчет людей может производиться непосредственно вручную преподавателем, однако, при большом количестве людей в аудитории это не представляется практически простым для реализации действием. Для таких случаев был разработан инструмент автоматического вычисления этой контрольной метрики – алгоритм подсчёта числа присутствующих на основе нейросетевого анализа фотографии.

Вопрос автоматического распознавания присутствующих решается с использованием моделей компьютерного зрения, в частности, сверточных нейронных сетей, эффективность которых при детекции объектов подтверждается многими исследованиями [135]. В рамках исследования были протестированы два подхода: детекция по телу и детекция по лицу. Для реализации этих подходов использовались предобученные нейросетевые модели YOLOv5 и MTCNN.

YOLOv5 [163], натренированная на датасете COCO, охватывает 80 категорий

объектов, включая класс “person”. Благодаря архитектуре однопроходной обработки изображений, данная модель демонстрирует высокую скорость и достаточную точность, превосходя в ряде случаев многопроходные аналоги [12]. Для повышения релевантности результатов модель была ограничена в распознавании исключительно объектов класса “person”.

Модель MTCNN [112], в отличие от YOLOv5, специализирована на детекции лиц и предварительно обучена на обширных выборках, что обеспечивает ей высокую точность (около 99% accuracy). MTCNN устойчива к частичному перекрытию лиц и плохим условиям съёмки. Данная нейросеть создана специально для обнаружения лиц и их ключевых точек (глаза, нос, уголки рта), а благодаря встроенному масштабированию исходного изображения она находит и маленькие лица вдалеке, и большие лица близко. В условиях реального использования, таких как аудитории с размещением студентов за партами или за компьютерной техникой, модель, ориентированная на распознавание лиц, показала лучшие результаты. Так, YOLOv5 демонстрировала недостаточную чувствительность, особенно при попытке зафиксировать студентов, находящихся на задних рядах. В то же время MTCNN успешно справлялась с задачей даже при частичном перекрытии лица, Рисунок 4.



Рисунок 4 – Пример работы модели MTCNN по распознаванию количества людей на фотографии

2.2 Модуль обнаружения заимствований в студенческих решениях

2.2.1 Анализ существующих подходов по выявлению заимствований в образовательных системах

Цифровизация ИТ-образования делает особенно актуальной задачу оценки самостоятельности программных решений студентов, поскольку массовое копирование кода снижает эффективность традиционных методов проверки заданий [113; 62]. Во многих случаях студенты заимствуют не отдельные идеи алгоритмов, а готовые фрагменты или целые реализации программ. При этом код обычно изменяется лишь формально: переименовываются переменные, меняется порядок строк, добавляются комментарии или незначительные конструкции. Подобные изменения позволяют скрыть факт копирования при поверхностной проверке, хотя не требуют от студентов глубокого понимания задачи. Исследования показывают, что различные формы заимствования при выполнении учебных заданий использовали от 50% до 79% студентов [65; 128]. Во время пандемии и массового перехода на дистанционное обучение масштабы проблемы дополнительно увеличились [64]. В настоящее время ситуация усложняется распространением систем автоматической генерации кода на основе генеративного искусственного интеллекта [42].

Существующие методы обнаружения плагиата в программном коде обычно разделяют на два основных класса: агрегированные и структурные. Агрегированные методы рассматривают программный код как набор отдельных фрагментов без учёта их последовательности и логических связей. К данной группе относятся векторные модели представления текста, включая word2vec [70] и doc2vec [101], методы частотного анализа, такие как BOW [164] и TF-ISF [133], а также подходы машинного обучения, основанные на контекстном и стилистическом сравнении программных решений [42; 90]. Основным преимуществом таких методов является высокая вычислительная эффективность и устойчивость к синтаксическим изменениям. Однако они плохо выявляют структурные преобразования программного кода, не учитывают особенности

конкретных языков программирования и часто дают результаты, которые сложно интерпретировать. Кроме того, подобные системы остаются чувствительными к намеренным изменениям кода, например к добавлению «шумовых» элементов или искусственных конструкций, снижающих степень сходства.

Структурные методы ориентированы на анализ синтаксической структуры и логических связей внутри программного кода. Одним из наиболее известных представителей данного класса является алгоритм MOSS (Measure Of Software Similarity) [137], разработанный в 1997 году и широко применяемый в образовательных учреждениях США. Его распространённость объясняется высокой точностью, устойчивостью к ложным срабатываниям и эффективностью при обработке большого количества программных решений. В основе работы MOSS лежит процедура нормализации исходного кода, в ходе которой удаляются пробелы, комментарии и изменённые идентификаторы, заменяемые стандартизированными обозначениями. После этого код разбивается на последовательности токенов (k-граммы), для которых вычисляются хеш-значения. Для сокращения объёма обрабатываемых данных и повышения устойчивости к незначительным изменениям применяется алгоритм winnowing, отбирающий подмножество хешей, называемых отпечатками, которые представляют документ. Кроме того, алгоритм не зависит от конкретной синтаксической структуры программы и эффективно справляется с такими трансформациями, как переименование переменных, замена логических блоков или перестановка операторов. Метод отпечатков, лежащий в основе MOSS, применим как к символьным n-граммам, так и к токенизированной программе. В частности, существует модификация алгоритма, при которой в процессе хеширования учитываются соседние токены переменной, что позволяет повысить точность выявления совпадений за счёт дополнительного контекста [111]. Другими популярными структурными алгоритмами также являются JPlag [129] и Sherlock [97]. Структурные методы также имеют ряд уязвимостей, например, они могут быть обойдены с помощью автоматизированных средств мутации кода, таких как фреймворк MOSSad [69], который внедряет семантически нейтральные

модификации, нарушающие структуру сравнения без изменения логики программы. Также к числу структурных методов можно отнести графовое представление текстовой информации, например, алгоритм GPLAG, который осуществляет построение графов с последующей оценкой степени их покрытия [82]. Итоговый сравнительный анализ всех классов методов приведен в таблице, Таблица 2.

Таблица 2 – Сравнительный анализ методов обнаружения заимствований в программном коде

Класс методов	Принцип работы	Преимущества	Недостатки	Примеры
Агрегированные	Представление кода как набора без учета структуры.	Вычислительная эффективность, устойчивость к простым заменам.	Низкая чувствительность к структуре, уязвимость к шумовым правкам.	Word2vec, TF-ISF, BOW
Структурные	Анализ синтаксического дерева или последовательности токенов.	Высокая точность, устойчивость к переименованиям и перестановкам.	Вычислительная сложность, уязвимость к семантически нейтральным мутациям (MOSSad).	MOSS, JPlag, GPLAG

2.2.2 Алгоритм обнаружения заимствований, учитывающий синтаксические особенности языков программирования

Описанные методы измеряют сходство текстов в целом, но не учитывают особенности блочной архитектуры исходного кода, где каждая структурная единица выполняет самостоятельную функцию и, при копировании, как правило, переносится без нарушения внутренней целостности. Исходя из вышеописанных проблем, необходима разработка такого алгоритма выявления заимствований, который бы сочетал в себе преимущества структурных подходов с возможностью интерпретации результатов и устойчивостью к типовым стратегиям обхода. Разработанный метод реализует блочную модель сопоставления кода, который основан на поиске наибольших совпадающих блоков кода в различных языках программирования.

С учётом распространённых способов обхода алгоритмов обнаружения заимствований, включающих переименование идентификаторов, изменение

структуры операторов, замену выражений на семантически эквивалентные и различные стилистические модификации, сначала производится предварительная очистка исходного текста от комментариев и незначимых пробелов, а также унификацию имён переменных, функций и других идентификаторов. Дополнительно выполняется нормализация выражений вида $++i$, $i++$ и $i += 1$, которые при синтаксических различиях в большинстве учебных задач обладают одинаковой семантикой.

Одним из важных этапов является разбиение исходного текста программы на строки, рассматриваемые как минимальные синтаксические единицы. Такая сегментация имеет принципиальное значение, поскольку многие языки программирования допускают запись программы в одну строку без нарушения её корректности. После предобработки программа преобразуется в унифицированное представление в виде массива строк, каждая из которых соответствует отдельной синтаксической единице. Далее выполняется сопоставление из первого файла с набором строк из второго файла для поиска совпадений. Если строка с номером i из первого файла похожа на строку j из второго файла согласно выбранной метрике $sim(l_i, l_j)$ и степень сходства превышает установленное пороговое значение τ , сравнивается следующая пара строк $i + 1$ и $j + 1$. Если они тоже похожи, то создается начальный фрагмент потенциально заимствованного блока. Далее аналогично рассматривается следующая пара строк, и это происходит, пока степень сходства превышает пороговое значение. Таким образом, алгоритм ориентирован не на поиск отдельных случайных совпадений строк, которые могут встречаться в различных частях программы, а на выявление крупных непрерывных блоков заимствованного кода.

Каждая согласованная пара блоков фиксированной длины k регистрируется как кандидат на совпадение. Для неё вычисляется усреднённая величина вероятности заимствования p по всем строкам блока, после чего происходит ранжирование найденных кандидатов. В приоритетном порядке рассматриваются те кандидаты, которые представляют собой более протяжённые блоки. Если длина блоков одинакова, предпочтение отдаётся тем, чья усреднённая вероятность

плагиата выше. Описанный алгоритм поиска заимствованных блоков кода может быть представлен в виде следующей схемы, Рисунок 5.

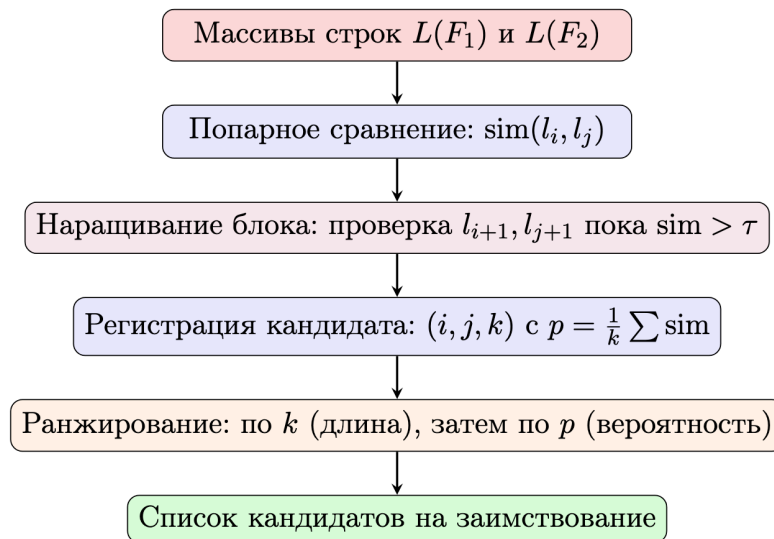


Рисунок 5 – Алгоритм поиска блоков заимствований в коде

Разметка и финальное сопоставление блоков между файлами выполняются с учётом конфликта пересечений и требований к однозначному покрытию. Кандидаты из отсортированного списка добавляются в итоговое множество по мере убывания приоритета. Если блок не пересекается с ранее занятыми строками, он включается в результат, при наличии конфликта он исключается, и, при возможности, вместо него рассматриваются альтернативные, более короткие, но непересекающиеся блоки. Важно подчеркнуть, что принятие решения о включении блока происходит не только на этапе формирования наиболее длинного совпадения, но и в процессе итеративного рассмотрения более коротких, но подходящих альтернатив, что позволяет добиться максимального охвата без нарушений логики соответствий. Пример работы алгоритма разрешения конфликтов между кандидатами представлен на рисунке, Рисунок 6.

В рассматриваемом случае, когда процесс доходит до кандидата, обозначенного красным цветом, в файле 1 соответствующий фрагмент ещё не выделен. Однако в файле 2 относящиеся к нему строки пересекаются с ранее размеченным зелёным блоком: строка с индексом j принадлежит одновременно

обоим блокам. При возникновении такого конфликта красный блок исключается из дальнейшего анализа, и вместо него рассматривается следующий по порядку кандидат, обладающий меньшим размером. Если к моменту обработки жёлтого блока соответственные ему строки в файле 2 не окажутся связанными с кандидатами, расположенными в очереди между жёлтым и красным блоками, то жёлтые блоки включаются в корректное соответствие и фиксируются в итоговой разметке.

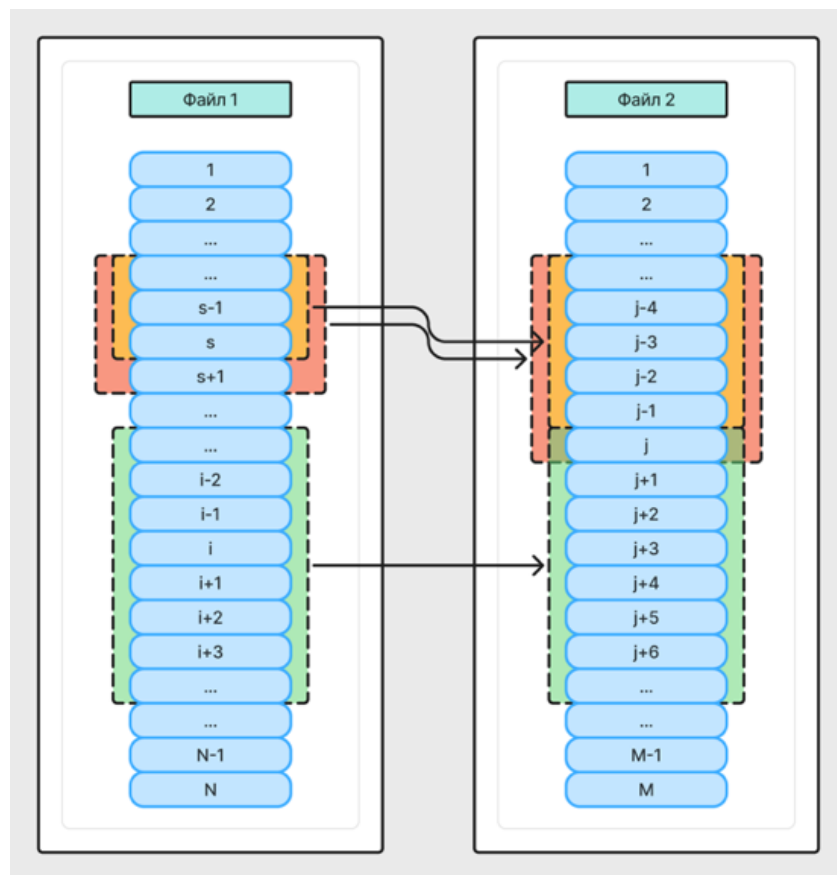


Рисунок 6 – Пример работы алгоритма разрешения конфликтов между кандидатами

Финальной стадией является отображение размеченных строк на исходный текст программы, в результате чего преподавателю предоставляется отчёт с вероятностной оценкой для каждой строки, поданной на вход системы. Это обеспечивает прозрачность и обоснованность выявленного заимствования, позволяя преподавателю сформировать взвешенное заключение о характере

нарушения. На этом этапе особенно важна корректность выбранной метрики – показателя, определяющего степень сходства между строками. Наилучшие результаты в рамках проведённого тестирования были получены при использовании расстояния Левенштейна [102], которое измеряет минимальное количество элементарных операций (вставки, удаления, замены), необходимых для преобразования одной строки в другую. Такая метрика хорошо справляется как с полными, так и с частичными совпадениями и устойчива к незначительным модификациям. Пример работы алгоритма представлен на рисунке, Рисунок 7, серым цветом подсвечен шаблон, красным цветом подсвечены совпадающие строчки, желтым цветом подсвечены похожие строчки.

1. Студент (25.05.2026, 20:21:29)
Вероятность 98.41%

solution.c

```

11 double a0;
12
13 if ((fin != NULL) && (fscanf(fin, "%lf", &a0) == 1)) {
14     {
15         fprintf(fout, "%lf", mss(fin, a0));
16     }
17
18     fclose(fin);
19     fclose(fout);
20     return 0;
21 }
22
23 double mss(FILE *fin, double a0)
24 {
25     double a, msf=a0, meh=a0;
26
27     while(fscanf(fin, "%lf", &a)==1){
28         if(meh+a>a) meh=meh+a;
29         else meh=a;
30
31         if(msf<meh)
32             msf=meh;
33     }
34     return msf;
35 }

```

solution.c

```

18 fclose(fin);
19 fclose(fout);
20 return 0;
21 }
22
23 double mss(FILE *fin, double a0)
24 {
25     double a, current_sum = a0, max_sum = a0;
26
27     while (fscanf(fin, "%lf", &a) == 1) {
28         // Текущая сумма - либо продолжаем пос
29         if (current_sum + a > a) {
30             current_sum = current_sum + a;
31         } else {
32             current_sum = a;
33         }
34
35         // Обновляем максимальную сумму
36         if (current_sum > max_sum) {
37             max_sum = current_sum;
38         }
39     }
40
41     return max_sum;
42 }

```

Рисунок 7 – Пример работы системы плагиата

2.2.3 Использование модуля обнаружения заимствований в различных образовательных сценариях

Одной из первых задач при анализе заимствований является унификация форматов и фильтрация исходных данных. Для программных решений обучающихся, хранящихся в базе цифровой платформы в виде архивов, требуется

их конкатенация в единый текстовый файл. При этом необходимо удалить фрагменты, совпадающие с исходными шаблонами, предоставленных преподавателем для помощи студентам, которые студенты не могут удалить из своей программы. Эти фрагменты кода являются общими для всех студентов и чаще всего содержат общий, технический код, который никак не влияет на алгоритмическую составляющую задачи. Его удаление позволяет избежать ложноположительных срабатываний, связанных с совпадениями в неизменяемых и общих частях кода решений студентов.

В предложенной системе студент не может копировать код ни из внешних источников, ни от других студентов. При попытке списывания ему приходится полностью перепечатывать каждую версию решения вручную. При этом неизбежно возникают опечатки, исправить которые без понимания кода он не сможет. Это многократно увеличивает трудоёмкость списывания и делает его практически неэффективным.

Представленный алгоритм детектирования плагиата можно применять не только для обнаружения списывания между студенческими решениями, но и для семантической проверки решений студентов на предмет совпадения с эталонным решением преподавателя. Помимо этого, решения, которые имеют большой процент «плагиата» с решением преподавателя, не проверяются на плагиат и исключаются из базы источников плагиата, поскольку этот большой процент совпадения показывает, что студент реализовал решение именно так, как предполагал преподаватель. Таким образом, в рассмотрение системы плагиата попадают только необычно реализованные решения, которые впоследствии были заимствованы другими студентами, что в итоге также помогает избежать ложноположительных срабатываний. Сравнение решения студента с эталонным решением также может быстро показать преподавателю, где различаются их решения, и оперативно предоставить студенту помощь в поиске ошибки. При этом совпадающие строчки в данном случае являются правильными, поэтому подсвечиваются зеленым цветом, а не красным, Рисунок 8.

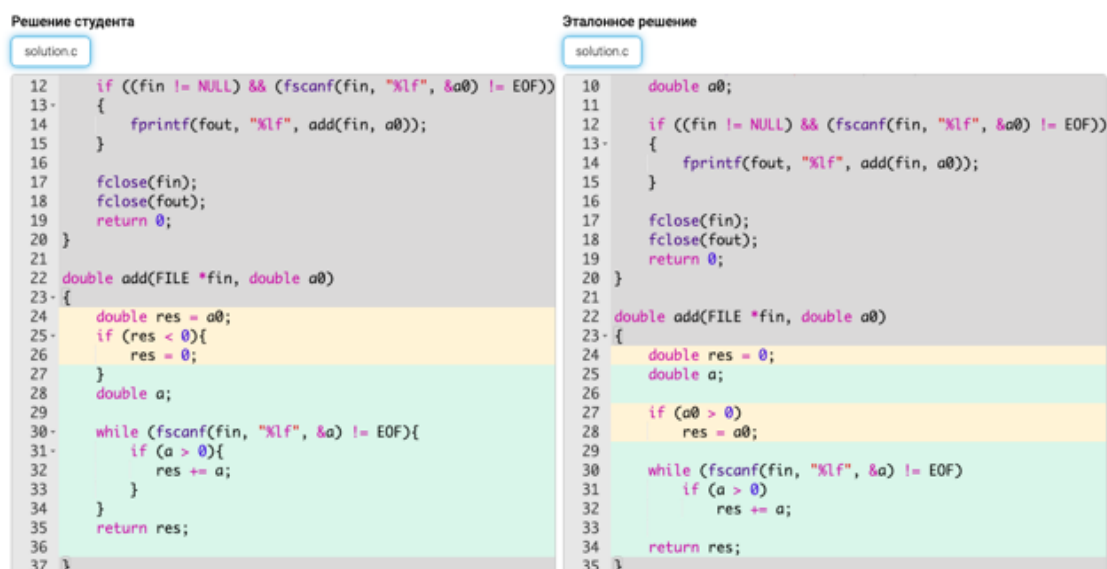


Рисунок 8 – Применение системы плагиата для сравнения решения студента с эталонным решением преподавателя

Для обнаружения заимствований целесообразно также использовать расширенные метаданные по каждой задаче студента: суммарное время решения задачи, количество потраченных на решение попыток, история изменения кода решений – всё это может использоваться для контекстного анализа потенциального заимствования. Так, например, массовая отправка решений с минимальным временным интервалом до окончания дедлайна, сопровождающаяся радикальным изменением стиля кода от попытки к попытке, то есть отсутствием высокого результата плагиата решения из одной попытки на решение из предыдущей попытки, может служить косвенным индикатором копирования. Помимо этого, например, падение времени, потраченного студентом на выполнение заданий при одновременном повышении уровня обнаруженного плагиата, может свидетельствовать о смене стратегии поведения – от самостоятельного решения в начале конкурса к попыткам быстрого копирования решений ближе к его завершению, Рисунок 9. Анализ подобных паттернов, основанных на поведенческой аналитике, позволяет значительно расширить функциональные возможности системы антиплагиата.

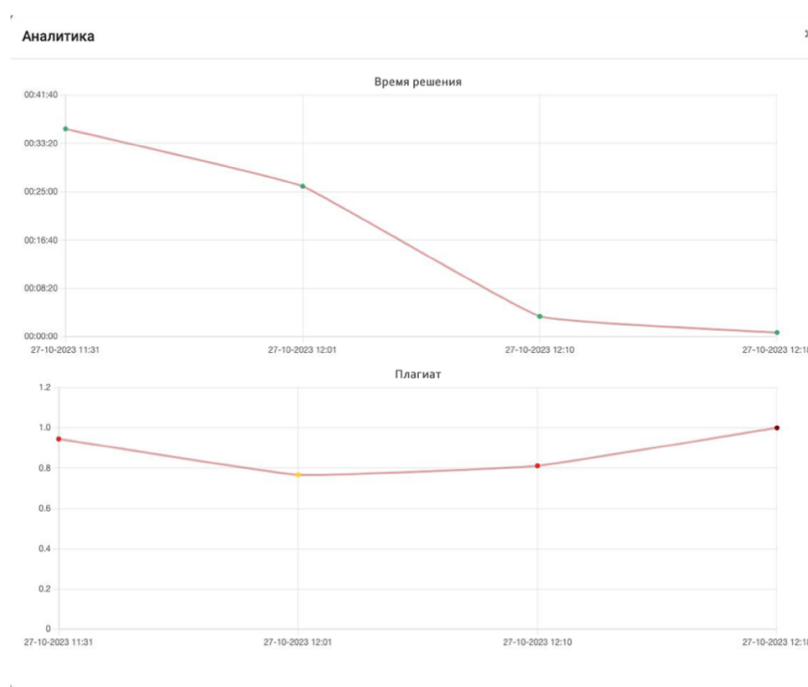


Рисунок 9 – Падение потраченного на решение задачи времени к концу контрольной с повышением уровня плагиата у студента

2.3 Модуль семантической проверки свободных текстовых ответов обучающихся

Значительная часть учебных заданий предполагает открытые формы ответов и пошаговое решение задач, что ограничивает применимость традиционных методов автоматической проверки и повышает нагрузку на преподавателей. В этой связи одной из ключевых задач интеллектуальной автоматизации образовательного процесса является разработка методов автоматической проверки развернутых текстовых ответов, которые будут обеспечивать интерпретируемые результаты автоматической оценки, снижать рутинную нагрузку на преподавателей, а также повысят надежность, объективность и прозрачность процесса оценивания большого количества работ студентов.

В работе предложен модуль автоматизированной проверки ответов студентов на открытые текстовые задания. В основе модуля лежит синтаксический разбор предложений с построением деревьев зависимостей, что позволяет анализировать структуры высказываний и учитывать семантическую полноту ответа, фокусируясь на содержании, а не на грамматической точности и

поверхностном сходстве.

Тестовые задания, предлагаемые обучающимся, можно разделить на два типа – открытые и закрытые. К закрытым относятся вопросы, в которых обучающийся выбирает один или несколько вариантов ответа из имеющихся. Открытые задания предполагают, что обучающийся формулирует собственный развернутый ответ. Существенным недостатком тестов с закрытыми вопросами является то, что они фокусируются преимущественно на способности воспроизводить изученную информацию, но не выявляют характера рассуждений или глубины осмысления темы со стороны студента. Открытые вопросы являются более содержательной формой для оценки знаний, поскольку они требуют осмысленного анализа информации и формулирования аргументированного ответа.

2.3.1 Анализ существующих методов автоматической проверки открытых текстовых ответов

В результате проведенного анализа автоматизированных систем проверки открытых текстовых ответов можно выделить четыре ключевых класса методов и подходов.

Первый класс методов опирается на технологии извлечения информации, относящиеся к направлению информационного поиска и предназначенные для преобразования неструктурированных текстовых данных в структурированный формат. Один из первых алгоритмов, предназначенных для автоматической проверки открытых ответов на английском языке с помощью извлечения информации, был описан в [155]. Метод заключается в построении графовых моделей, отражающих структуру ответа обучающегося и эталонного ответа преподавателя, после чего производится извлечение информации и ее последующее сопоставление. Отдельно следует выделить исследование, представленное в работе [16], в котором предложена система, используемая для оценки ответов посредством сравнения фактов, выделенных из текстовых данных на основе определенных грамматических правил. В работе [77] рассматривается методика сопоставления графов, сформированных на этапе извлечения

информации. Классические методы представления текста в графовой форме рассмотрены в работе [122].

Второй класс методов опирается на сравнительный анализ концепций. Этот подход предполагает выделение перечня основных понятий (минимальных концепций) из ответов студентов и эталонных решений, после чего проводится их сопоставление. Данный принцип лег в основу решения, описанного в работе [52]. В указанной системе с использованием фреймворка Generalised Phrase Structure Grammar формализуются синтаксические и семантические особенности языка, после чего ответы студентов и преподавателя преобразуются в наборы минимальных ключевых понятий. На следующем этапе производится количественное сравнение выделенных концепций, оценка ответа формируется с учетом весовых коэффициентов для каждой обнаруженной концепции.

Третий класс методов составляет применение для проверки текстовых ответов лингвистических корпусов – обширных подборок текстов, предназначенных для языкового анализа. В исследовании [79] рассматриваются классические алгоритмы на основе корпусов, позволяющие оценивать степень сходства между текстами. Данный способ базируется на формировании N-грамм слов. В работе [124] анализ схожести также основывается на оценке того, насколько последовательности N-грамм из эталонного ответа совпадают с фрагментами ответа обучающегося. Исследование в работе [126] также использует корпусный подход, однако фокусируется на организации синонимических колец для слов, что позволяет определять минимальное расстояние между ними, после чего измерение сходства между предложениями осуществляется путем сравнения их семантических векторных представлений.

Четвертым классом, который в последнее время стремительно набирает популярность, являются системы, опирающиеся на методы машинного обучения. Обычно такие технологии предполагают перевод слов в векторные пространства для оценки степени текстового сходства. В исследовании [147] описывается способ, базирующийся на разнообразных инструментах и техниках обработки естественного языка, включая SCBOW [98] – модель для предсказания слова по его

контексту, Wordnet [108] – лексическую базу данных, связывающую слова по смыслу, и Word2vec [72] – алгоритм получения векторных представлений слов с помощью нейронной сети.

В задачах обработки естественного языка применяются языковые модели на базе рекуррентных нейронных сетей и архитектуры трансформеров, которые способны «понимать» тексты на семантическом уровне, генерировать релевантные ответы на вопросы пользователей и поддерживать диалоговый режим взаимодействия. Несмотря на впечатляющие результаты больших языковых моделей, в академическом и профессиональном сообществе они подвергаются основательной критике [18], поэтому вопрос применения языковых моделей в областях с высокой чувствительностью к ошибкам остается спорным. Помимо этого, использование технологий больших языковых моделей в чистом виде не обеспечивает интерпретируемые результаты оценки.

Дополнительной трудностью выступает специфика синтаксической структуры русского языка: минимальные изменения порядка слов иногда порождают прямо противоположные по смыслу фразы, а модели, основанные на машинном обучении, могут не уловить подобные изменения. Для решения подобных задач, в частности, в работе [106], рассмотрен подход, в котором используются графовые сверточные нейронные сети совместно с синтаксическими деревьями зависимостей, что позволяет формировать семантическое векторное представление слов в предложениях.

В ряде исследований встречается интеграция различных подходов, описанных выше. Так, в работе [117] реализована объединённая модель, в которой применяется как граф знаний, отражающий взаимосвязи между понятиями, так и лингвистические корпуса значительного объёма. В исследовании [49] представлена система, сочетающая векторные малоразмерные представления с архитектурой Siamese Manhattan LSTM (MaLSTM) [123], её эффективность была протестирована на примерах реальных, открытых студенческих ответов в рамках цифровой образовательной платформы. К разряду комплексных решений можно отнести модель, описанную в [107], где одновременно используются методы сегментации

текста, морфологического анализа, построения семантической матрицы и синтаксического графа. В работе [84] рассмотрен алгоритм, основывающийся на идентификации ключевых терминов через анализ графлетов. Анализ подходов к построению систем автоматизированной проверки также показал, что из всех рассмотренных решений лишь система, представленная в [16], ориентирована на обработку текстов на русском языке. Итоговый сравнительный анализ всех классов методов приведен в таблице, Таблица 3.

Таблица 3 – Сравнительный анализ классов автоматизированных систем проверки открытых текстовых ответов

Класс методов	Принцип работы	Преимущества	Недостатки
Извлечение информации	Построение графовых моделей ответа и эталона с последующим сопоставлением фактов	Интерпретируемость результатов, работа со структурой текста	Зависимость от качества грамматических правил, сложность настройки
Сравнительный анализ концепций	Выделение ключевых понятий и их количественное сопоставление с весовыми коэффициентами	Простота идеи, прозрачность оценки	Трудоёмкость составления полного перечня концепций
Лингвистические корпуса	Оценка сходства через N-граммы, синонимические кольца и семантические векторы на базе корпусов	Не требует вручную заданных правил	Зависимость от объёма корпуса, слабый учёт порядка слов
Машинное обучение	Векторизация слов (Word2vec, SCBOW) и применение нейросетей, включая LLM	Высокая точность на больших данных, учёт семантики	Неинтерпретируемость (LLM), «галлюцинации», чувствительность к малым изменениям в русском языке
Интегративные подходы	Комбинирование графов знаний, корпусов, векторных моделей и синтаксического анализа	Компенсация недостатков отдельных методов	Сложность балансировки компонентов, низкая интерпретируемость итоговой оценки, высокая ресурсоёмкость

2.3.2 Лингвистический процессор и построение графов синтаксического разбора

В ходе исследования была поставлена и решена задача по созданию системы автоматизированной проверки развернутых ответов на открытые вопросы, которая будет рассчитана на работу с русскоязычными текстами. Преподавателю

необходимо заранее подготовить совокупность коротких эталонных предложений, репрезентирующих ключевые идеи задания. Ответы студентов могут отличаться по сложности и структуре. Автоматическая проверка проводится методом сравнения ответов обучающихся с эталоном, при этом анализируются не только точные совпадения слов, но и синонимы, антонимы, различные грамматические формы, формы склонений, альтернативные написания собственных имен и дат, а также местоимения. Система распознаёт многоуровневые отрицания, оценивая их влияние на общий смысл. За искажение смысла посредством антонимов и отрицаний выставляются штрафные баллы.

Для классификации результатов используются два пороговых значения. Если оценка ниже нижнего порога – ответ признаётся неверным, если выше верхнего – считается зачтённым. В случае спорных оценок между порогами требуется ручная проверка преподавателем, которому поступает соответствующее уведомление. Задача решается с использованием методов информационного поиска, которые совмещают построение графов разбора с сопоставлением фактов. Такой подход основывается на синтаксическом анализе текста, позволяющем получать графы синтаксического разбора. Далее из этих графов формируются деревья, которые затем сравниваются между собой.

Синтаксический анализ – это процесс сопоставления последовательности лексем формальной грамматике языка. Для программной реализации данной задачи используются лингвистические процессоры, выполняющие морфологический и синтаксический анализы текста. Результат их работы – граф зависимостей, отражающий синтаксическую структуру предложения, пригодный для последующего семантического анализа, Рисунок 10.

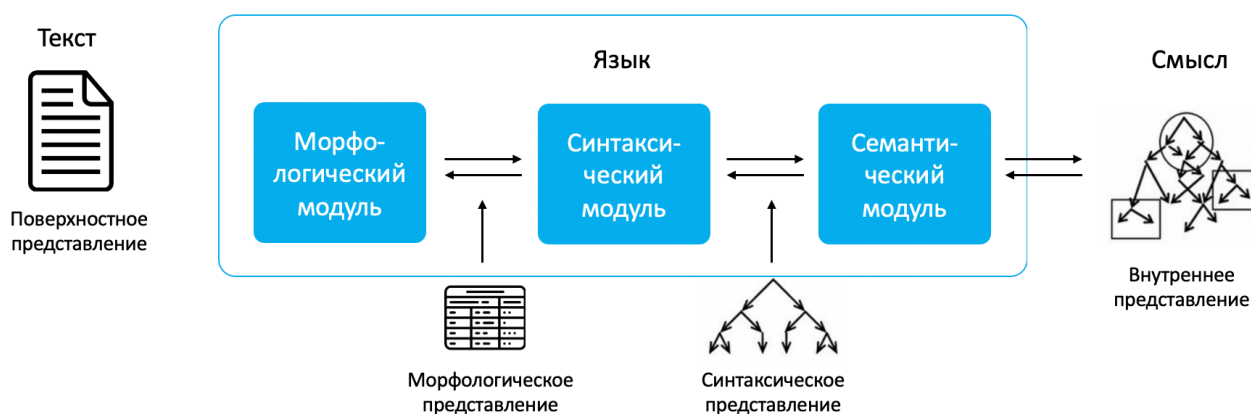


Рисунок 10 – Схема анализа текста лингвистическим процессором

В реализации лингвистических процессоров применяются два подхода: основанный на правилах и основанный на машинном обучении с учителем. Первый предполагает ручное формулирование лингвистических правил, необходимых для синтаксической разметки. Например, в Томита-парсере эти правила составляются командой лингвистов, однако заранее определить их полный набор невозможно, что приводит к затруднениям при разборе сложных предложений [16].

Второй подход, основанный на методах машинного обучения, предусматривает обучение моделей на размеченных корпусах. Одним из успешных примеров реализованных моделей для русского языка является проект Natasha [115], который включает в себя предобученные эмбединги слов (Naves), а также программные инструменты для решения всех базовых задач обработки естественного языка: разделение текста на предложения (Razdel), морфологический и синтаксический анализ (Slovnet), лемматизация (Pymorphy) и извлечение именованных сущностей и дат (Yargy). Эти модели позволили собрать гибкий и эффективный лингвистический процессор, не требующий дополнительных действий от преподавателя по сравнению с инструментами, использующими метод, основанный на правилах. Для работы с иностранными текстами можно использовать аналогичный инструмент CoreNLP [61]. Результатом работы соответствующего процессора является граф синтаксического разбора, Рисунок 11, где связи между словами определяются согласно межъязыковой типологии Universal Dependency Relations [156].

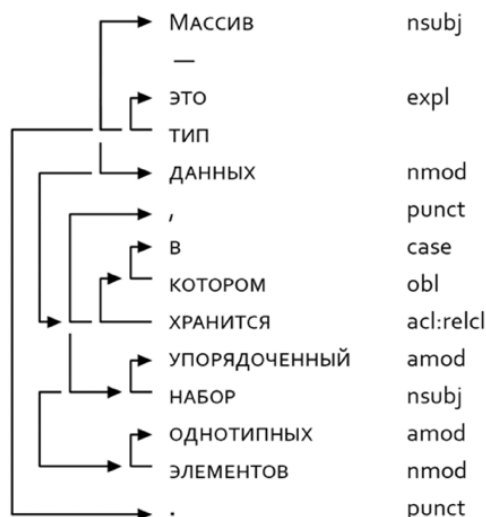


Рисунок 11 – Построенное лингвистическим процессором графа синтаксического разбора

2.3.3 Предобработка синтаксических графов и обработка отрицаний

Сравнение синтаксических графов предложений на русском языке представляет собой сложную вычислительную задачу из-за свободного порядка слов, возможности полного изменения смысла предложения от перестановки нескольких слов местами, наличия циклов и нескольких компонент связности в полученном синтаксическом графе. Таким образом, даже у похожих предложений могут быть абсолютно разные графы синтаксического разбора. Для повышения эффективности алгоритмов сравнения требуется предварительная обработка графов: удаление несущественных для смысла частей речи и слов, удаление циклов и объединение различных компонент связности в одну для приведения графов к виду деревьев, на которых алгоритмы сравнения работают существенно быстрее. Для этих целей применяется библиотека для работы с графами NetworkX [116].

Решение задачи удаления циклов основано на приоритизации лексически значимых частей речи: глаголов, существительных, имен собственных и местоимений. Ребро, входящее в наиболее значимый узел, подлежит удалению в первую очередь. В случае невозможности такого выбора удаляется последняя взаимосвязь при возврате в исходную вершину, что гарантирует устранение всех циклов. После этого, в случае обнаружения нескольких компонент связности, то

есть отдельных деревьев, они объединяются в одно путем выбора главного корня и наследовании от него остальных корней. Если у студента части сложного предложения будут указаны в другом порядке, то направление ребра наследования между корнями может быть перенаправлено алгоритмом без понижения вероятности засчитывания ответа.

Работа описанного алгоритма предобработки синтаксических графов показана на примере предложения: «Движения макроскопических тел при скорости, много меньшей скорости света». На рисунке, Рисунок 12, сверху изображён исходный граф, а внизу – дерево, являющееся результатом после его предобработки, которая включает разрыв циклов, объединение двух компонент связности и удаление лексически малозначимых частей речи.

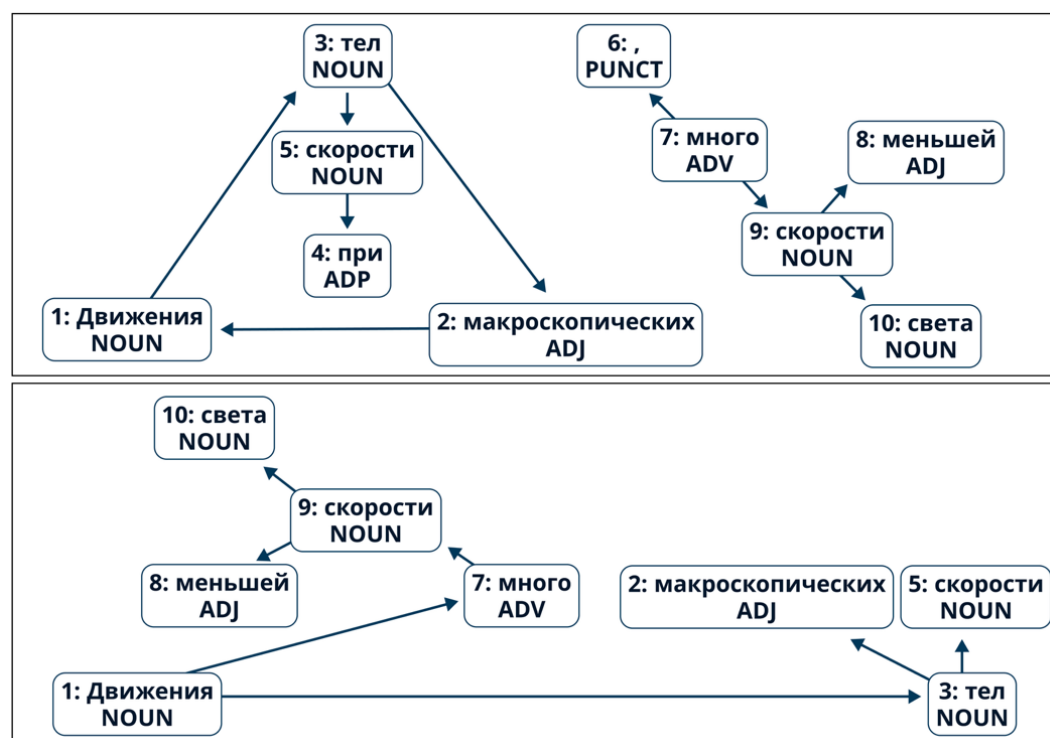


Рисунок 12 – Предобработка графа синтаксического разбора

Одной из важнейших составляющих алгоритма является представление и обработка слов с отрицательным значением. Отрицание может относиться либо к главному слову (например, в выражении «совсем не яркий» слово «не» является потомком слова «яркий»), либо к подчинённому слову (например, в выражении

«нет места» слово «нет» становится главным по отношению к «месту»). Если отрицание идёт от подчинённого слова к главному, то отрицательный элемент удаляется, а его наследники становятся наследниками его родителя, после чего родительскому элементу присваивается отметка, что он несёт отрицательный смысл. В противоположной ситуации, когда отрицание идёт от родителя к наследникам, наследники получают отметку отрицательного смысла без удаления родительского элемента, поскольку он чаще всего является лексически значимой частью речи. Результат работы алгоритма обработки отрицаний представлен на рисунке, Рисунок 13, где изображён граф, построенный по следующему предложению: «Я не мог не согласиться, что нет никаких противоречий».

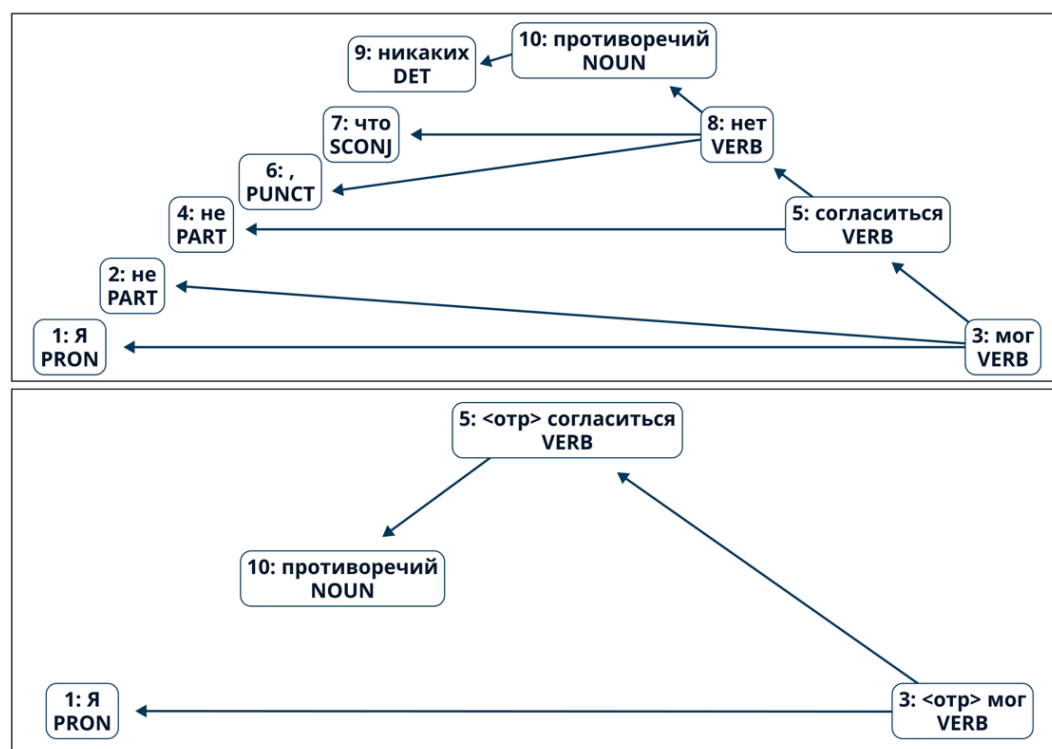


Рисунок 13 – Обработка отрицаний в синтаксическом дереве. Исходный граф и полученное дерево с проставленными метками отрицательного смысла

2.3.4 Сравнение синтаксических деревьев и вычисление итоговой метрики

После построения синтаксического дерева требуется определить выделенные типы узлов для их корректного сравнения между собой. Первый тип – «Имена», путём предварительной обработки все элементы, относящиеся к имени,

объединяются в один узел и сохраняются в универсальном формате. При сравнении таких узлов учитываются частичные совпадения, например, если в одном из узлов отсутствует отчество, но остальные компоненты совпадают, узлы считаются также совпадающими («А.С. Пушкин», «Александр Пушкин», «Пушкин»). Второй тип – «Даты»: аналогично обработке имён, все части одной даты объединяются в единый узел, и совпадением дат считается частичное совпадение компонент их узлов («12 декабря 2023», «12.12.2023», «12 декабря»). Работа сравнения этих двух типов узлов показана на примере пары следующих предложений: «С сентября 1830 года Михаил Юрьевич Лермонтов учился в Московском университете» и «С 01.09.1830 М. Ю. Лермонтов учился в Московском университете», Рисунок 14. Здесь имена и даты, несмотря на различия в написании, группируются в отдельные узлы и будут определены совпадающими.

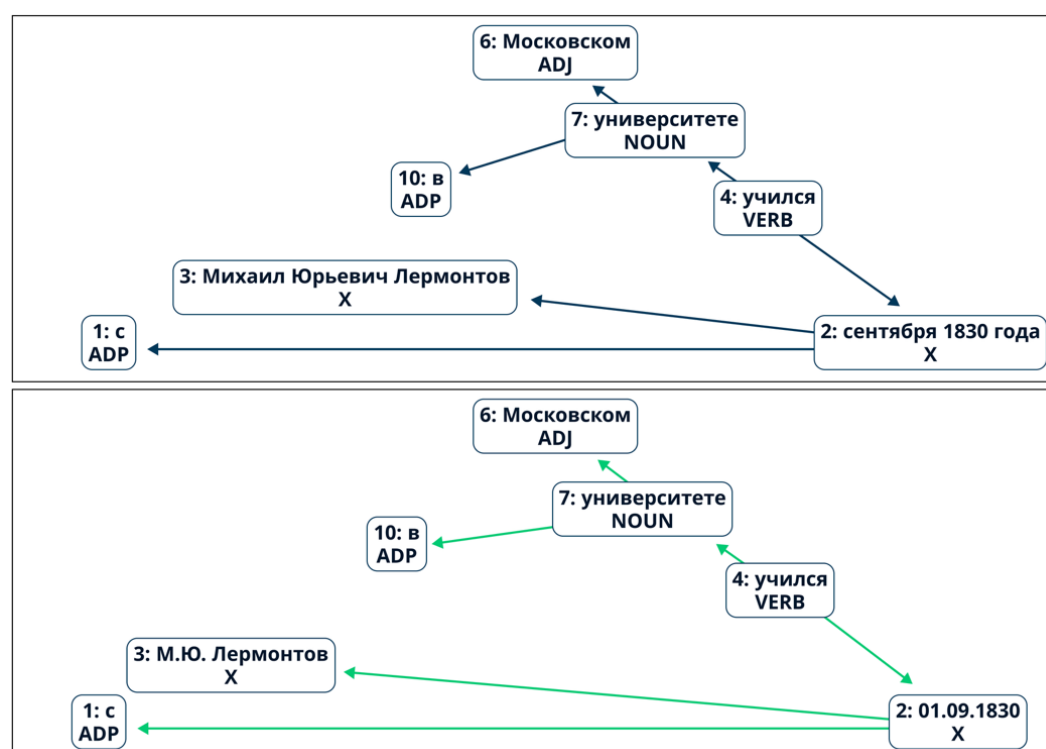


Рисунок 14 – Объединение и сравнение узлов типа «Имена» и «Даты»

К третьему типу «Текст» относятся все остальные вершины. Узлы этого типа проходят предобработку последовательными преобразованиями, такими как лемматизация (приведение к нормальной форме) и стемминг (приведение к основе

слова, то есть удаление окончаний). Для описания алгоритма их сравнения введем отношение сравнения этих типов узлов. Узлы являются совпадающими, если после преобразований они идентичны или являются синонимами и их метки отрицательности совпадают, или если они являются антонимами и метка отрицательности есть только у одного из этих узлов. Узлы являются противоположными, если после преобразований они идентичны или являются синонимами и метка отрицательности есть только у одного из этих узлов, или если они являются антонимами и их метки отрицательности совпадают. Узлы являются наложившимися друг на друга, если они являются совпадающими или противоположными. Для идентификации синонимов и антонимов используется открытый репозиторий, структурированный в виде графа [136]. Для обработки местоимений устанавливается род всех текстовых узлов, что определяется до этапа стемминга. После этого слова этих узлов будут являться идентичными, если хотя бы одно слово является местоимением и род этих слов совпадает. Итоговый алгоритм сравнения различных узлов представлен на схеме, Рисунок 15.

Непосредственное сравнение графов реализуется с помощью алгоритма поиска оптимального пути между графами, описанного в работе [44]. С его помощью можно искать оптимальные наложения одного графа на другой с помощью элементарных преобразований, например, добавление или удаление новых вершин или ребер. В алгоритм передается функция для сравнения вершин, в данном случае это функция определения наложившихся узлов, описанная выше. После получения оптимального наложения вершин проводится анализ наложившихся пар узлов, среди них отдельно выделяются совпадающие и противоположные узлы, а также те, что оказались наложенными в ходе работы алгоритма или остались без пары: для них можно провести отдельную кросс-проверку для поиска дополнительных совпадающих узлов.

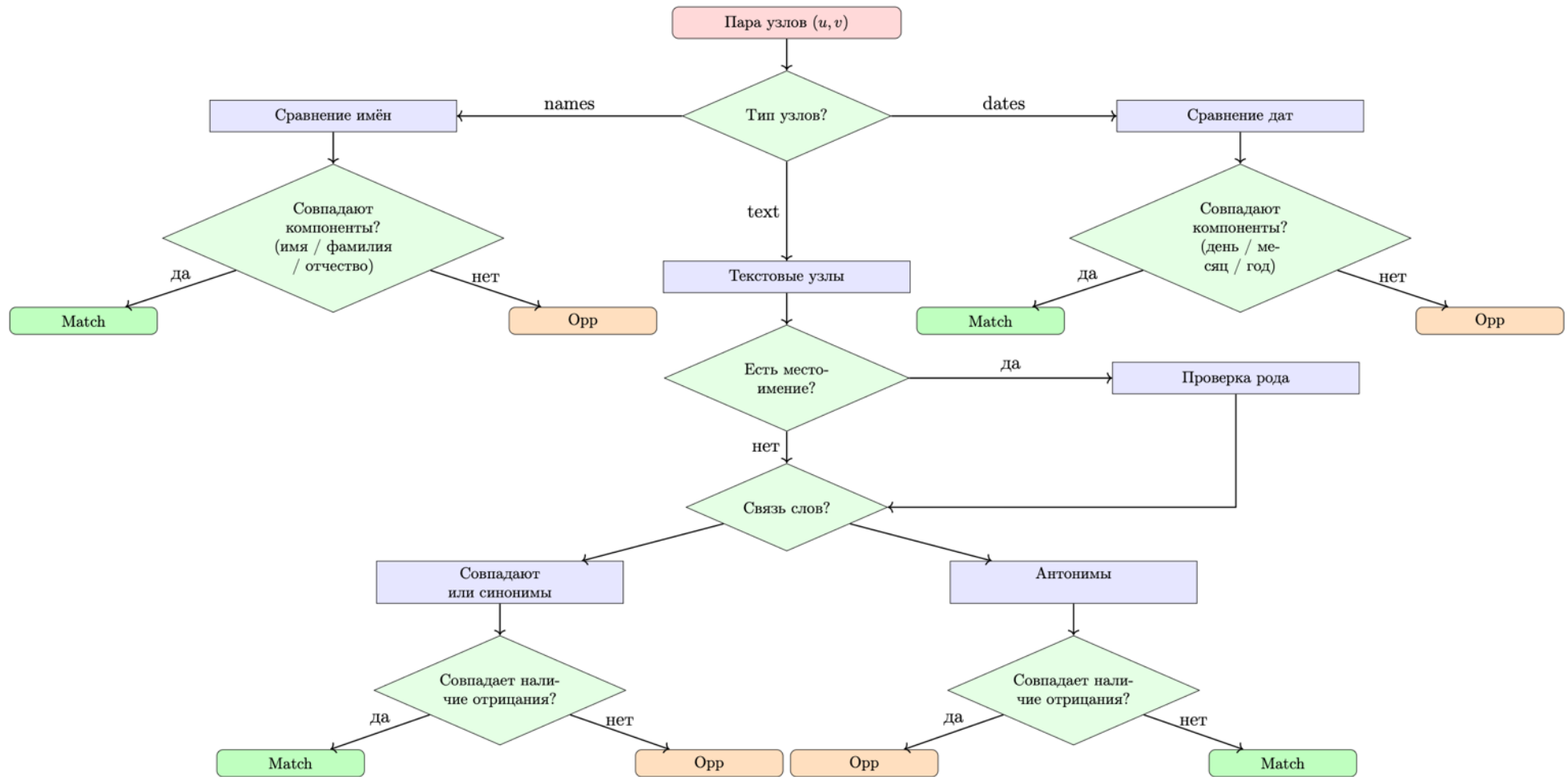


Рисунок 15 – Схема алгоритма сравнения узлов различных типов

Противоположные узлы проходят дополнительную постобработку с целью удаления многоуровневых отрицаний: при рекурсивной обработке критерием распознавания многоуровневого отрицания является наличие у пары противоположных узлов наследников, также являющихся противоположными узлами. После этого пары противоположных узлов удаляются по двое, таким образом, остаётся либо одна пара противоположных узлов, если глубина рекурсии была нечётной, либо ни одной, если она была чётной. Работа данного алгоритма удаления двойных отрицаний представлена на рисунке, Рисунок 16, на примере двух предложений: «Я не мог не согласиться с этим утверждением» и «Я мог согласиться с этим утверждением». Пары противоположных узлов и их противоположных наследников после обработки выглядят так: «мог (отрицательно) – мог» и «согласиться (отрицательно) – согласиться». После удаления таких многоуровневых отрицаний они исключаются из подсчета итоговой вероятности.

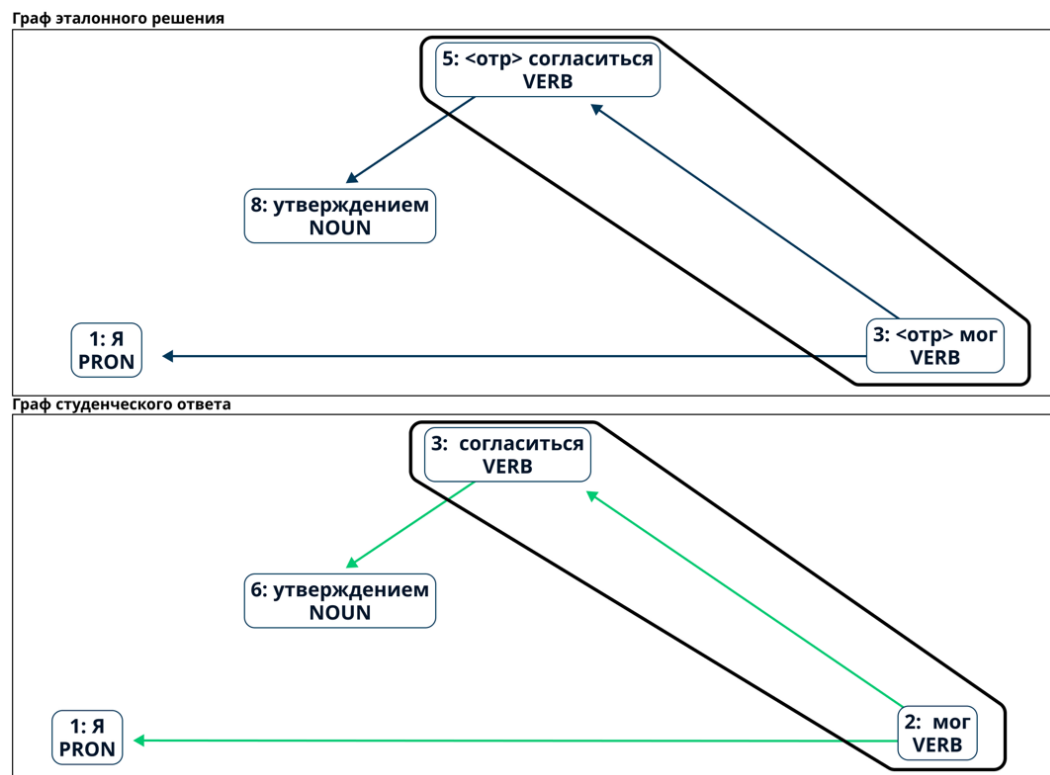


Рисунок 16 – Рекурсивный поиск многоуровневых отрицаний, чёрным цветом обведены найденные совпадающие цепочки

При подсчете итоговой метрики необходимо учитывать несколько ключевых метрик: доля наложения, измеряющая процент наложенных вершин, включая наложенные вершины после проведения кросс-проверки, доля совпадения, учитывающая только совпадающие вершины, а также количество полученных противоположных вершин, которое говорит об итоговых логических ошибках в решении обучающегося и влияет на итоговый результат значительными штрафами.

Итоговый пример работы алгоритма показан на следующих двух предложениях: «Изображение не изменится, фото будет менее ярким» и «Картинка не поменяется, оно будет тусклым», Рисунок 17. Пары слов «изображение» и «картинка», «изменится» и «поменяется», являются синонимами, поэтому они являются совпадающими узлами. Слова «фото» и «оно» также являются совпадающими узлами, поскольку у них одинаковый средний род, а одно из них является местоимением. Узлы «ярким (отриц.)» и «тусклым» являются совпадающими, так как «ярким» и «тусклым» являются антонимами и у этих узлов различаются метки отрицательности. Таким образом, доля наложения и доля совпадения этих предложений будет равна 100%.

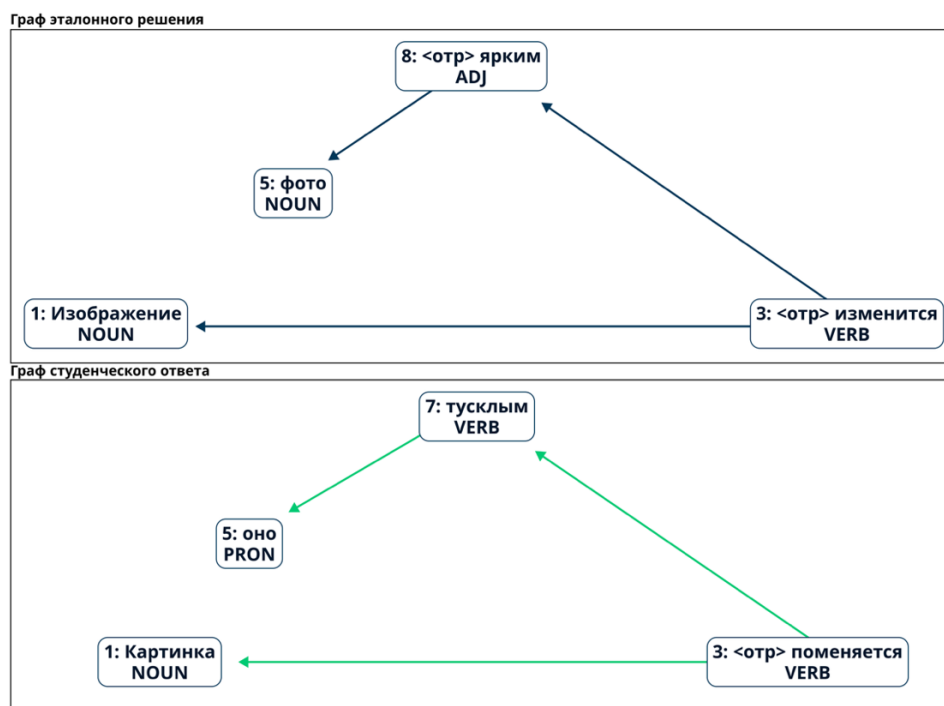


Рисунок 17 – Пример работы алгоритма сравнения синтаксических деревьев, учитывающий синонимы, антонимы, местоимения и отрицания

Особое значение имеет возможность параметрической настройки: преподаватель может гибко задавать пороговые значения метрик, варьировать систему штрафов и поощрений, адаптируя процесс под конкретные задания. При этом необходимо подчеркнуть необходимость индивидуальной подстройки этих параметров к каждому заданию с учетом статистических характеристик, поскольку универсальные значения оказываются неэффективными в условиях вариативности содержания и структуры заданий. Эффективность системы также во многом зависит от качества эталонных решений и формулировок заданий, что делает актуальной задачу разработки инструментов для поддержки преподавателя при их создании.

2.4 Нейросетевые ассистенты на основе контекстно-ограниченного взаимодействия в образовательном процессе

Интеллектуальная автоматизация в образовании реализуется в том числе через использование виртуальных ассистентов и нейросетевых диалоговых систем. Такие решения позволяют предоставлять обучающимся пояснения, подсказки и навигацию по учебному материалу, а преподавателям – помощь при подготовке заданий, формировании контента и анализе результатов обучения, при этом сохраняя учебный контекст и не подменяя самостоятельную деятельность студентов. Виртуальные ассистенты интегрируются в цифровую образовательную платформу и ориентируются на конкретные учебные сценарии, анализируя действия пользователей, контекст выполнения заданий и результаты обучения. Формируемые в процессе их использования данные могут дополнительно применяться для образовательной аналитики и построения индивидуальных образовательных траекторий.

2.4.1 Анализ существующих подходов к разработке интеллектуальных диалоговых систем

Традиционно использование чат-ботов активно развивается в коммерческих секторах – в банковской сфере, онлайн-торговле, клиентском обслуживании и

технической поддержке [15; 23]. В последние годы подобные подходы начинают постепенно адаптироваться к образовательным задачам, в том числе через интеграцию в мессенджеры и цифровые образовательные платформы учебных заведений. Использование ботов в образовательном процессе может использоваться для создания аналогов поисковых систем по материалам конкретного курса или вуза в целом, работающих в режиме диалога [5], а также проведения анкетирования и информирования студентов [36].

Однако внедрение чат-ботов именно в качестве педагогических ассистентов всё ещё находится на стадии апробации, что обусловлено этическими и методологическими сложностями. Данная проблема стала еще более актуальной после активного распространения больших языковых моделей (LLM). Несмотря на открывшиеся перспективы автоматизации образовательного процесса, например, помощь в создании адаптивных учебных планов, интеллектуальную поддержку студентов в процессе обучения и автоматизацию при создании проверки заданий, остаются актуальными этические проблемы их применения: плагиат и нарушение академической честности, генерация ложной информации, конфиденциальность данных [66]. Использование LLM в образовании усугубляется восприимчивостью обучающихся к информации, которую предоставляют языковые модели, и в случае «галлюцинаций» нейросети студенты могут воспринять это за корректную информацию. Поэтому, несмотря на опасения экспертов, что у студентов снизится уровень критического мышления, в реальности в связи с необходимостью постоянной проверки ответов языковых моделей ситуация обратная. Данные проблемы можно решить, если использовать преимущества больших языковых моделей для поиска по собственным базам знаний курса, тем самым объединяя нейросетевые подходы с традиционными системами поиска информации, что реализовано, например, в чат-боте ChatEd [160]. Это решает проблему неточной информации, предоставляемой нейросетевыми ботами, поскольку языковые модели выступают в данном случае только для сохранения истории поиска и обработки пользовательских запросов, в ответ система присылает точные ссылки на загруженные преподавателем материалы, однако, в целом бот сильно зависит от

качества загруженных в него материалов. Некоторые другие этические и психологические проблемы при создании образовательного помощника решены в рамках разработки бота EduChat [71], предобученного на учебниках по психологии для оказания эмоциональной поддержки и использующего метод диалога, стимулирующего самостоятельное мышление.

2.4.2 Предметно-ориентированный чат-бот с заданной областью знаний

В педагогической практике возникают ситуации, при которых необходимо строго ограничить предметную область, предоставляемую ботом [1]. Основной задачей являлась разработка образовательного чат-бота, который предоставлял бы студентам круглосуточный доступ к учебным материалам и ответам на типовые вопросы, при этом сохраняя контроль преподавателя над содержанием предметной области и исключая выдачу готовых решений заданий. Система должна была поддерживать обработку запросов на естественном языке, включая неточные формулировки, разговорные выражения и языковые отклонения, а также обеспечивать сбор статистики взаимодействий для анализа учебной активности и поддержки персонализированных образовательных траекторий.

В отличие от коммерческих чат-ботов, образовательные системы предъявляют ряд дополнительных требований к архитектуре и функциональности. В частности, необходимо учитывать ограниченный объём обучающих данных на начальном этапе разработки [148], поддерживать широкий спектр вариантов формулировки вопросов и обеспечивать возможность ведения свободного диалога [39]. В качестве основы для реализации был выбран фреймворк Rasa [37; 121], являющийся открытой платформой для построения диалоговых систем с использованием методов машинного обучения и не требующий больших объёмов тренировочных данных.

В процессе исследования и разработки чат-бота была сформирована предметно-ориентированная база знаний, включающая сведения о структуре курса, типовых вопросах и ответах, а также используемой терминологии. Помимо этого, была реализована возможность расширения языковой модели за счёт обработки

разговорных и нестандартных выражений, что особенно важно при взаимодействии со студенческой аудиторией. Апробация системы на базе учебных групп механико-математического факультета МГУ имени М.В. Ломоносова показала, что уже на начальных этапах использования чат-бот способен заметно сократить количество повторяющихся обращений к преподавателю. По мере накопления истории взаимодействий система стала точнее определять тематику запросов, расширяя набор поддерживаемых диалоговых сценариев и повышая релевантность ответов. При этом студенты использовали чат-бот не только для решения учебных задач, но и для неформального общения. Это потребовало внедрения диалогов общего характера. Для решения данной задачи была предложена модель разделения языковых и семантических компонентов, позволяющая создавать предметные надстройки поверх базовых коммуникативных шаблонов. Такой подход упрощает адаптацию чат-бота к новым дисциплинам и сводит участие преподавателя преимущественно к подготовке предметной базы знаний в текстовом виде. Пример взаимодействия студента с ботом, демонстрирующий возможности построения многоуровневых сценариев и ведения диалога, представлен на рисунке, Рисунок 18.

- Привет
- Привет, Иван! Как ты там без меня?
- Плохо(
- Иван, что произошло??
- Проблемы с массивами
- Присылаю тебе файл, в котором представлен пример программы, основанной на массивах. Тебе помогло это?
- Нет(
- Присылаю тебе файл, в котором представлен пример программы с зачета с двумя массивами.
- Спасибо!
- Всегда к твоим услугам!

Рисунок 18 – Пример диалога с чат-ботом

2.4.3 Использование больших языковых моделей в образовательных системах

Рост вычислительных возможностей способствовал активному внедрению больших языковых моделей (LLM) в образовательные системы, что позволило автоматизировать педагогические процессы, повысить уровень персонализации

обучения и оптимизировать организацию учебного процесса [103; 142; 50].

Выбор больших языковых моделей для использования в образовательной среде представляет собой задачу со множеством критериев, включающую оценку производительности модели, требований к вычислительным ресурсам, качества генерируемого контента, возможностей адаптации к различным предметным областям, а также вопросов безопасности и совокупной стоимости эксплуатации. Существенное значение имеют вычислительная сложность модели, размер контекстного окна и объем необходимых аппаратных ресурсов, поскольку многие образовательные организации не располагают высокопроизводительной серверной инфраструктурой. В связи с этим практическое применение находят либо облегченные локальные модели в квантизованных версиях, снижающие требования к видеопамяти, либо облачные API-решения, позволяющие отказаться от собственной вычислительной инфраструктуры. Локальное развертывание обеспечивает больший контроль над моделью и персональными данными, тогда как облачный подход уменьшает инфраструктурные ограничения, но создает зависимость от внешнего провайдера и риски передачи данных третьим сторонам, вследствие чего на практике нередко используется гибридная архитектура, при которой чувствительные операции выполняются локально, а ресурсоемкие задачи делегируются в облако. Дополнительно выбор модели определяется спецификой образовательных задач: модели, ориентированные на программный код, более эффективны в инженерных и программных дисциплинах, тогда как для работы с русскоязычными текстами предпочтительны решения, обученные на крупных русскоязычных корпусах, что обуславливает применение комбинированных архитектур из нескольких специализированных моделей.

2.4.4 Нейросетевой ассистент обучающегося для контекстной индивидуальной поддержки

В рамках данного исследования реализован нейросетевой ассистент, интегрированный в архитектуру цифровой образовательной платформы и функционирующий как специализированный сервис педагогической поддержки.

Его основная функция заключается в генерации педагогически корректных подсказок, направленных на самостоятельное выявление и исправление ошибок обучающимся без предоставления готового решения.

При массовом обучении традиционная модель взаимодействия между студентом и преподавателем перестает масштабироваться, что затрудняет оперативную обработку большого количества типовых ошибок. Дополнительно проблема усугубляется тем, что студенты часто избегают обращений за помощью по базовым вопросам, вследствие чего простые синтаксические и логические ошибки могут надолго блокировать выполнение задания. При этом стандартные системы автоматической проверки обычно ограничиваются выдачей только результата, без предоставления механизмов анализа и интерпретации ошибок.

Ранние работы в области автоматизации педагогической поддержки были сосредоточены на разработке интеллектуальных обучающих систем, основанных на формализованных моделях знаний и заранее заданных правилах [59]. В классических исследованиях, посвященных когнитивным наставникам, показано, что подобные системы способны эффективно сопровождать обучающегося при решении задач, приближаясь по результативности к индивидуальному обучению с преподавателем [157]. Вместе с тем данные подходы характеризуются высокой стоимостью разработки и слабой масштабируемостью, что существенно ограничивает их применение в условиях современных цифровых образовательных платформ с большим количеством разнородных заданий.

Дальнейшее развитие исследований привело к появлению методов автоматической генерации обратной связи на основе анализа программного кода обучающихся, в рамках которых были предложены алгоритмы, использующие синтаксические деревья и статический анализ для выявления типовых ошибок [99]. Эти решения позволили частично автоматизировать проверку и сократить время реакции системы, однако их педагогическая ценность оказалась ограниченной [94]. Формируемые сообщения, как правило, ориентированы на разработчиков и не всегда понятны начинающим обучающимся. В результате такие инструменты эффективно указывают на наличие ошибки, но не способствуют формированию

целостного понимания причин ее возникновения.

С появлением больших языковых моделей фокус исследований сместился в сторону генеративных подходов к поддержке обучения программированию. В ряде работ показано, что языковые модели способны объяснять фрагменты кода, генерировать примеры и отвечать на вопросы обучающихся на естественном языке [151; 131]. Однако прямое использование открытых диалоговых интерфейсов в образовательном процессе сопровождается рядом рисков. К ним относятся галлюцинации и потеря контекста конкретной задачи, а также склонность моделей к выдаче готовых решений, которые могут не подходить под все требования [95]. Эти особенности противостоят требованиям академической честности и могут приводить к снижению когнитивной активности обучающихся [130].

Анализ литературы показывает, что эффективное использование генеративных языковых моделей в образовательных платформах требует гибридного подхода, сочетающего их возможности с архитектурными и дидактическими ограничениями учебного процесса. В отличие от внешних диалоговых интерфейсов, такие ограничения, а также формализованный контекст учебной задачи, автоматически извлекаются из внутреннего представления цифровой образовательной платформы и определяют фиксированный регламент взаимодействия ассистента с обучающимся.

В качестве базовой языковой модели в разработанном нейросетевом ассистенте используется открытая архитектура класса Mistral с числом параметров, обеспечивающим компромисс между качеством генерации и вычислительной эффективностью. Модель разворачивается в контролируемой серверной среде и не имеет прямого доступа к внешним сетевым ресурсам. Поведение модели задается системной инструкцией, жестко ограничивающей допустимый формат ответов. Ассистенту запрещено генерировать завершенные программные фрагменты, которые могли бы быть напрямую использованы как решение задачи. Вместо этого модель формирует пояснения, наводящие вопросы и краткие теоретические справки.

Ключевой особенностью предложенного решения является использование

механизма генерации с расширенным контекстом, формируемого на основе данных образовательной платформы. При формировании запроса к языковой модели учитываются условие задачи, форматы входных и выходных данных, программный код обучающегося, логи выполнения тестов и скрытое эталонное авторское решение.

Использование эталонного решения в расширенном контексте позволяет выявлять первопричины ошибок в случаях, когда диагностические сообщения компилятора или интерпретатора отражают вторичные дефекты. Наличие эталонного решения также обеспечивает формирование подсказок, направляющих обучающегося к академически корректному способу решения, предполагаемому преподавателем, и исключающих отклонения от методически заданной логики. Помимо этого, использование ассистентом эталонного решения повышает устойчивость генерации в задачах с высокой алгоритмической сложностью, в том числе в ситуациях, когда развернутая языковая модель без опоры на эталон не была бы способна корректно воспроизвести полное решение самостоятельно. Кроме того, эталонное решение выступает в роли семантического ориентира при работе с алгоритмическими языками и учебными диалектами, отсутствующими в обучающей выборке языковой модели, что расширяет область применимости ассистента в рамках специализированных образовательных курсов.

Взаимодействие с ассистентом инициируется через управляющую кнопку без возможности произвольного текстового ввода, после чего формируется формализованный запрос, содержащий исходный код и диагностическую информацию проверки. Такой подход снижает риски манипуляции языковой моделью и предотвращает нецелевое использование ассистента при наличии доступа к эталонному решению задачи. Работа системы организована в итеративном режиме: после каждой неудачной попытки отправки решения обучающийся может повторно обращаться к ассистенту, а генерируемые подсказки формируются с учетом текущего состояния кода и ранее исправленных ошибок.

В результате создается среда, в которой студент получает доступ к качественной интеллектуальной поддержке без необходимости вручную

формулировать или передавать модели полный контекст задачи, как это требуется при использовании открытых LLM моделей. Это не только упрощает взаимодействие, но и способствует формированию устойчивых навыков самостоятельного решения задач. В этом контексте LLM модели выступают не в роли инструмента для списывания, а как полноценный тьютор, ориентированный на развитие когнитивной активности обучающегося.

Алгоритм генерации подсказок адаптируется к типу выявленной ошибки. При наличии синтаксических ошибок или ошибок компиляции ассистент концентрируется на объяснении нарушенного правила языка и помогает локализовать источник проблемы. При этом учитывается, что сообщения компилятора не всегда указывают на фактическое место ошибки, а некоторые синтаксические нарушения могут не приводить к ошибке компиляции, но вызывать некорректное выполнение программы. Сравнение с эталонным решением позволяет выявлять подобные случаи и корректно интерпретировать причину сбоя, Рисунок 19. В результате обучающийся получает пояснение, ориентированное на понимание механизма работы языка, а не на механическое исправление конкретной строки с ошибкой. При обнаружении семантических ошибок в логике работы решения стратегия генерации подсказок меняется. В этих случаях ассистент обращает внимание обучающегося на расхождения между ожидаемой и фактической логикой программы, не указывая напрямую на конкретное исправление, Рисунок 20. В ответе помощника может содержаться вопрос, наталкивающий на самостоятельное исправление ошибки, или объяснение, почему текущее решение некорректно проходит тесты.

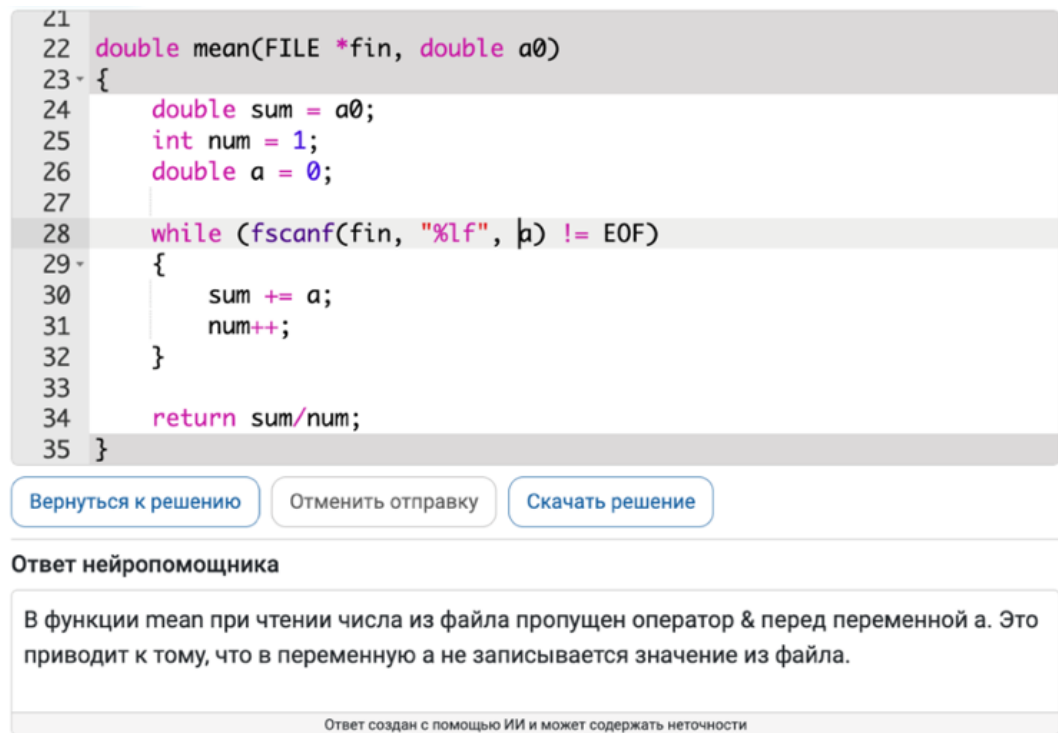


Рисунок 19 – Определение неявной синтаксической ошибки нейросетевым ассистентом

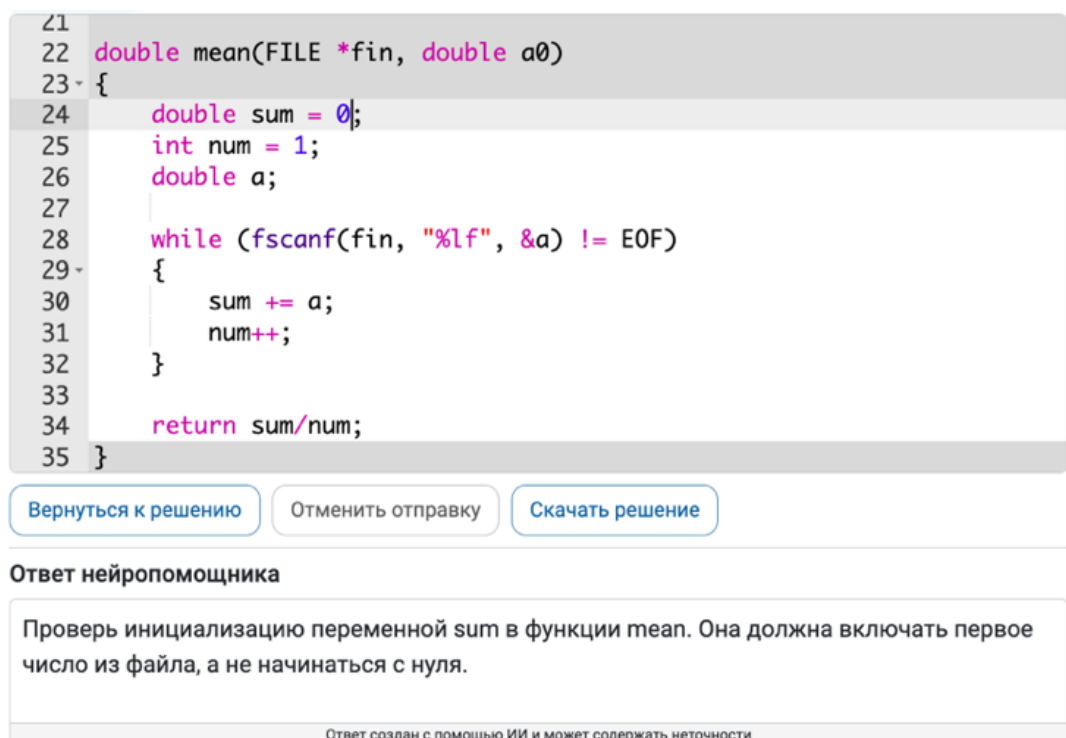


Рисунок 20 – Ответ нейросетевого ассистента в случае логической ошибки в программе

Для определения особенностей разработанного решения проведено сопоставление с нейросетевыми ассистентами, применяемыми в образовательных платформах. В качестве близких по назначению решений рассмотрены CodeGrade AI Assistant, Iris, интегрированный в платформу Artemis, и GenMentor. Результаты сравнения представлены в таблице, Таблица 4. Сравнение показывает, что существующие решения уже используют контекст учебной деятельности и интеграцию языковых моделей непосредственно в образовательные платформы. Основные различия связаны со способом организации взаимодействия: CodeGrade AI Assistant и Iris предоставляют диалоговый интерфейс, тогда как GenMentor ориентирован на персонализацию обучения на основе модели обучающегося и адаптивной траектории. Разработанный ассистент сочетает управляемый сценарий взаимодействия, автоматическое формирование контекста и использование локальной открытой языковой модели. При этом ограничения взаимодействия задаются не только инструкциями языковой модели, но и архитектурой образовательной платформы. Это позволяет использовать LLM как специализированный компонент педагогической поддержки, а не как самостоятельный универсальный диалоговый сервис.

Таблица 4 – Сравнение нейросетевых ассистентов в образовательных платформах

Характеристика	Разработанный ассистент	CodeGrade AI Assistant	Iris (Artemis)	GenMentor
Интеграция в образовательную платформу	+	+	+	+
Использование контекста учебной деятельности	+	+	+	+
Управляемый сценарий взаимодействия	+	частично	частично	частично
Ограничение выдачи готового решения	+	настраивается	+	не заявлено
Автоматическое формирование запроса	+	–	–	–
Персонализация поддержки	контекстная	частичная	контекстная	на основе модели обучающегося
Локальное использование открытых моделей	+	не заявлено	экспериментально	не заявлено

2.4.5 Нейросетевой ассистент преподавателя для создания образовательного контента и сопровождения занятий

Использование LLM-моделей позволяет снизить объём рутинной работы преподавателя за счёт автоматизации подготовки учебных материалов и генерации новых заданий. Создание задач выполняется строго на основе материалов, прикрепленных преподавателем, благодаря чему уменьшается вероятность появления нерелевантных или ошибочных ответов, связанных с «галлюцинациями» модели. Такой подход обеспечивает соответствие задания материалам занятия, позволяет учитывать узкоспециализированные темы курса, отсутствующие во внешних источниках данных, а также усложняет решение задач с помощью универсальных нейросетевых сервисов, обученных на общих наборах данных. Дополнительным преимуществом является полная интеграция генерации в инфраструктуру платформы: преподавателю достаточно инициировать процесс через интерфейс, после чего сбор контекста, генерация и публикация заданий выполняются автоматически.

Система также поддерживает генерацию задач, тематически и структурно близких к существующим, что повышает вариативность учебных материалов, снижает возможности списывания и способствует более объективной оценке знаний. В области обратной связи реализован ассистент преподавателя, анализирующий вопросы студентов и формирующий диагностические комментарии с описанием возможных причин ошибок и затруднений. Полученный результат предоставляется преподавателю в виде черновика, который может быть отредактирован и подтвержден перед отправкой студенту, что позволяет сократить время анализа при сохранении персонализированного характера обратной связи.

Интеграция LLM непосредственно в образовательную платформу устраняет ограничения при использовании внешних генеративных сервисов, где требуется вручную формировать сложные промпты, передавать контекст и переносить результаты в систему.

Выводы по главе 2

Во второй главе разработан комплекс интеллектуальных модулей, обеспечивающих автоматизацию ключевых процессов образовательной деятельности, включая учёт посещаемости, проверку заданий различных типов, а также поддержку участников образовательного процесса с использованием методов искусственного интеллекта. Рассмотрение данных задач позволило перейти от теоретически обоснованной необходимости интеграции интеллектуальных модулей к их практической реализации в рамках цифровой образовательной платформы.

В ходе исследования предложена архитектура и алгоритмическое обеспечение модуля учёта посещаемости, ориентированного на работу в условиях очного, дистанционного и гибридного обучения. Показано, что использование данных из различных источников цифрового следа позволяет формировать более объективные и верифицируемые показатели посещаемости по сравнению с традиционными методами. Введены контрольные метрики, использующие методы компьютерного зрения, которые обеспечивают проверку корректности получаемых данных.

Разработан модуль обнаружения заимствований в студенческих решениях, учитывающий синтаксические особенности языков программирования и устойчивый к типовым модификациям кода. Предложенный алгоритм блочного сопоставления позволяет учитывать структурные особенности программ, выявлять устойчивые совпадения и дополнять их поведенческой аналитикой.

Предложен и реализован модуль автоматизированной проверки семантической составляющей текстовых ответов обучающихся, основанный на построении и сравнении графов синтаксического разбора. Показано, что использование синтаксически ориентированного подхода к проверке ответов обеспечивает более устойчивые и интерпретируемые результаты по сравнению с методами, основанными исключительно на статистическом сходстве текстов.

Особое внимание уделено разработке нейросетевых ассистентов

обучающегося и преподавателя, функционирующих в контексте образовательного процесса. Рассматриваемые решения включают как ограниченные предметно-ориентированные диалоговые модули, предназначенные для ответов на частые вопросы по учебным материалам, так и контекстные интеллектуальные помощники, функционирующие с учётом текущего состояния учебной деятельности студента. Разработан нейросетевой ассистент преподавателя, предназначенный для поддержки процессов создания образовательного контента, подготовки заданий и сопровождения учебного процесса.

В результате показано, что разработанные интеллектуальные модули обладают универсальностью и могут быть интегрированы в различные цифровые образовательные среды как единая система взаимосвязанных компонентов, обеспечивающих автоматизацию ключевых этапов образовательного процесса.

ГЛАВА 3. МАТЕМАТИЧЕСКИЕ МЕТОДЫ И АЛГОРИТМЫ ДЛЯ ЗАДАЧ АНАЛИЗА ЦИФРОВОГО СЛЕДА ОБУЧАЮЩИХСЯ И УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ

3.1 Анализ существующих методов сбора и анализа цифрового следа в образовательных системах

С переходом к дистанционным и смешанным форматам обучения значительно возрастает объем данных, генерируемых в ходе образовательной деятельности, что расширяет возможности применения методов аналитики цифрового следа обучающихся для диагностики, прогнозирования результатов и проверки гипотез. Цифровой след, формируемый в процессе взаимодействия пользователя с образовательной платформой, представляет собой массив информации о действиях студента: посещаемость, попытки решений заданий, набранные баллы, параметры взаимодействия с учебным контентом и т.п.

Наиболее значимой особенностью анализа цифрового следа является построение моделей поведения обучающихся и использование этих моделей для персонализации образовательных траекторий, выявления затруднений и прогнозирования рисков академической неуспеваемости. Данный подход соответствует концепции data-driven образования [43] и принципам информационно-обоснованного принятия решений (Data-Driven Decision Making, DDDM), согласно которым сбор и анализ данных об образовательной активности студентов должны выполняться непрерывно и систематически. Это позволяет оперативно проверять гипотезы, вносить изменения в организацию учебного процесса и персонализировать обучение.

Современные цифровые образовательные платформы обеспечивают работу с различными типами данных, включая административную информацию, предпочтения студентов при выборе форматов обучения, особенности взаимодействия пользователей с платформой, а также накопленные данные об успеваемости. Дополнительно могут анализироваться социальные взаимодействия обучающихся, например коммуникация в чатах или участие в совместных

проектах, а также отдельные психофизиологические показатели.

Для хранения и обработки подобных данных используются специализированные системы, поддерживающие стандарт xAPI, предназначенный для унификации структуры хранения цифрового следа и обеспечения совместимости данных при передаче между различными образовательными системами. В рамках данного стандарта фиксируются так называемые statements – утверждения в формате JSON, описывающие действия пользователя, например: «Студент просмотрел видео». Эти данные сохраняются в специальное хранилище Learning Record Store (LRS) и далее могут анализироваться средствами бизнес-аналитики или специализированных дашбордов.

Среди популярных LRS-решений выделяется система SCORM Cloud LRS [138], известная как инструмент для тестирования SCORM-пакетов, которая также предоставляет базовые функции облачного LRS, включая просмотр стейтментов и экспорт данных в различных форматах. Более мощным решением является система Watershed LRS [161], ориентированная на крупные организации и имеющая расширенные аналитические возможности, включая конструктор дашбордов с множеством типов отчетов. Система Learning Locker [100] представляет собой open source LRS, которая предлагает базовый просмотр стейтментов и конструктор дашбордов, ориентированный на ответы к конкретным вопросам (например, анализ недельной активности). Ещё одним решением является система GrassBlade LRS [85], которая включает в себя детальный анализа взаимодействия студентов с различными мультимедийными материалами.

На современном этапе развития образовательных технологий персонализированные и адаптивные системы обучения получили широкое распространение практически на всех уровнях образования, включая начальную, основную и среднюю школу, а также высшее образование [110; 154; 149]. Подобные подходы применяются не только в рамках учебного процесса, но и при оценивании результатов обучения. Существующие персонализированные обучающие системы целесообразно классифицировать по используемым методологическим основаниям. Первую группу составляют решения, в которых

адаптация реализуется за счёт процедур персонализации образовательного контента. К данному классу относится система ALEKS [46], в которой стартовой точкой обучения выступает входное диагностическое тестирование. В дальнейшем образовательная траектория корректируется на основе результатов выполнения заданий, а при достижении заданного порога успешности обучаемому предлагается промежуточная проверка освоения материала. Менее жёстко формализованный механизм адаптации реализован в системе Knewton [60], где каждый последующий ответ студента используется для динамической корректировки образовательного процесса, включая выявление недостаточно усвоенных тем и организацию повторения ранее изученного материала.

Во вторую группу входят адаптивные системы, основанные на применении специализированных интеллектуальных технологий, ориентированных на анализ и закрепление учебного материала. Одним из ранних примеров является система SCHOLAR, использующая экспертную систему с набором правил для оценки корректности ответов обучающегося и выбора дальнейших шагов обучения [53]. Более современные решения, такие как DreamBox, опираются на методы машинного обучения [159]. При формировании последующих заданий в подобных системах учитывается не только правильность ответов, но и параметры процесса решения задач, включая скорость реакции и иные поведенческие характеристики.

К третьей группе относятся подходы к адаптивному обучению, основанные на анализе цифрового следа студента, формируемого в процессе его взаимодействия с образовательной средой. В рамках данных подходов применяются методы интеллектуального анализа данных, направленные на выявление закономерностей в образовательных данных и их использование для поддержки принятия педагогических решений [139; 134]. Практическая значимость анализа цифрового следа заключается в возможности диагностики учебных затруднений, прогнозирования результатов обучения и выявления характерных образовательных траекторий. Эмпирические исследования подтверждают эффективность таких подходов, демонстрируя высокую точность классификации и интерпретации характеристик обучающихся на основе

накопленных поведенческих данных [40]. В работе [80] показано, что использование данных цифрового следа обучающихся положительно влияет на успешность завершения образовательных курсов и позволяет повысить долю студентов, осваивающих учебный материал в установленные сроки. Эффективность персонализированного подхода подтверждена статистическим сравнением результатов контрольной и экспериментальной групп. Итоговый сравнительный анализ всех групп подходов приведен в таблице, Таблица 5.

Вместе с тем использование цифрового следа в адаптивных образовательных системах сопровождается рядом ограничений. Существенное место среди них занимают этические вопросы, связанные с определением прав собственности на образовательные данные, условиями их хранения после завершения обучения, а также защитой чувствительной информации обучающихся. Дополнительно остаётся дискуссионным вопрос интеграции внешних источников данных, отражающих неформальное и самостоятельное обучение вне рамок образовательной организации, в структуру цифрового следа студента.

Таблица 5 – Сравнительный анализ подходов к анализу цифрового следа и персонализации обучения

Группа подходов	Назначение	Механизм работы
LRS-решения (хранение)	Унифицированный сбор данных по стандарту xAPI	Регистрация JSON-утверждений о действиях студентов, построение дашбордов
Персонализация контента	Адаптация учебного материала под уровень студента	Входная диагностика, корректировка программы, промежуточный контроль
Интеллектуальные системы	Анализ и закрепление знаний с помощью ИИ	Экспертные правила и/или машинное обучение с учётом поведенческих характеристик
Анализ цифрового следа	Прогнозирование результатов и выявление рисков	Интеллектуальный анализ данных учебной активности, выявление закономерностей

3.2 Модуль сбора и отображения аналитики данных цифрового следа

В условиях цифровизации образовательного процесса возрастает значимость поведенческой аналитики цифрового следа как инструмента точной диагностики

учебных траекторий, поскольку традиционные метрики (баллы, посещаемость, заимствования) недостаточны для выявления глубинных различий в усвоении материала и индивидуальных поведенческих особенностей студентов [48]. Использование цифрового следа позволяет анализировать не только результаты, но и процесс взаимодействия с учебным контентом, выявлять поведенческие паттерны и динамику обучения. На основе таких данных возможно категориальное разделение обучающихся на группы с различной успеваемостью («отличники», «хорошисты», «двоечники») с характерными поведенческими профилями, включая вовлечённость, уровень заимствований и временные стратегии выполнения заданий. Анализ переходов между указанными категориями и изменений ключевых показателей, включая темпы набора баллов, количество попыток выполнения заданий, посещаемость и наличие заимствований, позволяет выявлять риски снижения успеваемости и строить прогностические модели для адаптации образовательного процесса и своевременной педагогической поддержки.

Использование аналитических моделей, основанных на анализе поведенческого цифрового следа, позволяет решать широкий круг задач, связанных с повышением качества обучения. К таким задачам относятся прогнозирование академических результатов, построение адаптивных образовательных траекторий, оценка уровня освоения материала, вовлечённости студентов и качества учебных заданий. Это даёт возможность своевременно выявлять проблемные элементы курса и корректировать образовательный контент [22]. Дополнительно анализ временных рядов поведения обучающихся позволяет сравнивать учебные группы, оценивать эффективность используемых педагогических подходов, выявлять студентов со стабильно высокими результатами, а также повышать точность систем обнаружения заимствований за счёт учёта нетипичных поведенческих шаблонов.

В рамках настоящего исследования были проанализированы цифровые следы студентов, проходивших обучение на механико-математическом факультете МГУ имени М.В. Ломоносова в течение двух лет. Данные были собраны с использованием разработанного модуля аналитики, который позволяет собирать,

агрегировать и визуализировать ключевые показатели учебной активности, помогая преподавателям анализировать успеваемость студентов, выявлять тенденции и формулировать гипотезы для улучшения учебного процесса. Собранные данные включают в себя несколько ключевых характеристик, которые обрабатываются и усредняются для курсов, тем и контестов.

К числу анализируемых показателей относятся: доля набранных баллов, посещаемость, среднее количество попыток решения задач, среднее время решения задач, а также доля задач, признанных списанными. Доля набранных баллов учитывает баллы, полученные студентом выполнение заданий, за посещаемость и за особые достижения, зафиксированные преподавателем. Общая сумма нормируется на установленную верхнюю границу, выставленную преподавателем как необходимую для выполнения контеста, что обеспечивает сопоставимость контестов с различной максимальной оценкой. Посещаемость фиксируется как флаг, указывающий на выполнение условий для её засчитывания. Среднее количество решений представляет собой среднее количество потраченных попыток по всем задачам контеста, к выполнению которых приступал студент. Среднее время решения по задачам рассчитывается исходя из суммы временных промежутков, потраченных на написание каждой попытки. Для выявления случаев списывания применяется встроенный модуль антиплагиата, способный определять заимствования как внутри одной группы, так и по всем предыдущим попыткам студентов других групп, проходящих этот курс. Критерием списывания служит эмпирически установленный порог вероятности плагиата – не менее 95%, при этом из анализа исключаются случаи, где решения студентов совпадают с эталонным решением на 100%, что бывает в задачах, которые предоставлены студенту с полным решением в шаблоне в качестве примеров.

Для визуализации данных модуль предлагает несколько типов графиков, которые можно настроить в зависимости от потребностей преподавателя. Временное шкалирование может быть выбрано для контестов, Рисунок 21, тематических модулей, Рисунок 22, или целиком курсов, что позволяет анализировать данные на разных уровнях детализации. Визуализация графиков,

охватывающих весь курс, представляет особую ценность при проведении сравнительного анализа учебных результатов различных групп.

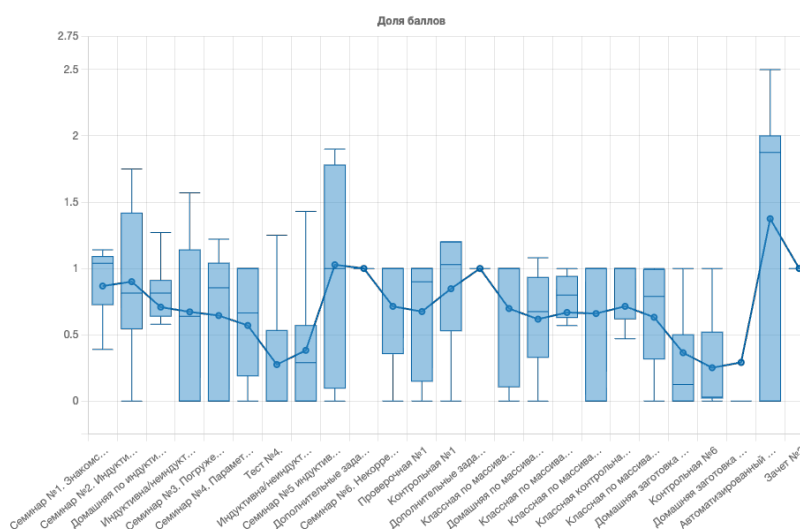


Рисунок 21 – Средняя доля баллов по группе в масштабе контестов

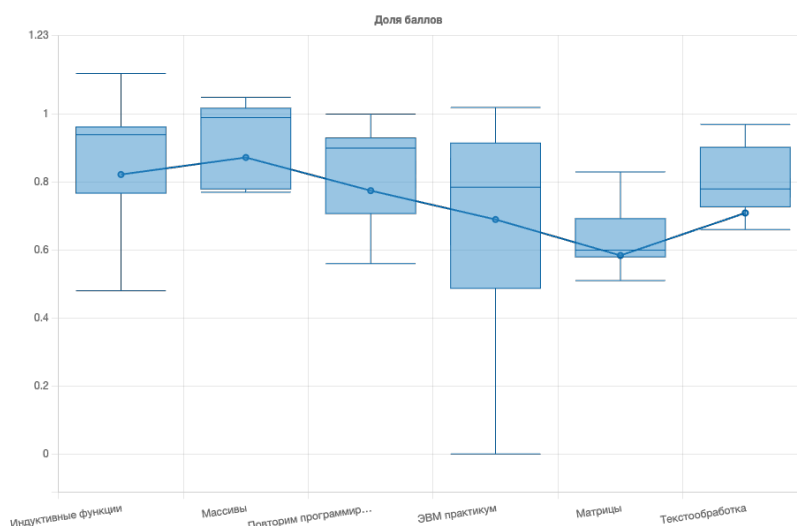


Рисунок 22 – Средняя доля баллов по группе в масштабе тем

На данных графиках отображаются усреднённые показатели в виде boxplot-графиков с линией, соединяющей средние значения, что рекомендуется использовать в больших группах для избежания перегруженности графиков. В случае небольшого количества студентов может быть выбрана индивидуализированная визуализация в виде линейных графиков, отображающих данные каждого студента отдельно, что способствует более детальному сравнительному анализу. На первом рисунке, Рисунок 23, представлен пример

сравнительного графика распределения долей набранных баллов двух студентов, а на втором рисунке, Рисунок 24, представлен сравнительный график их посещаемости, построенный по принципу накопительной суммы, что позволяет наглядно определить, в какой момент студент прекратил посещение занятий.

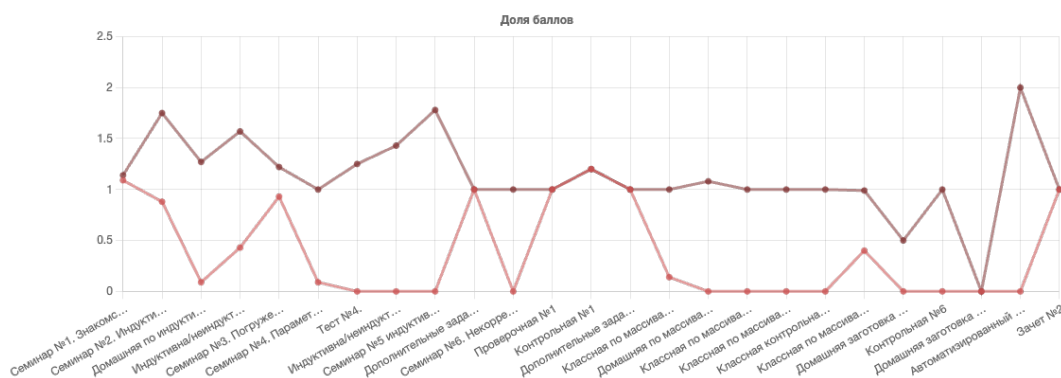


Рисунок 23 – Сравнение доли набранных баллов отдельно двух студентов

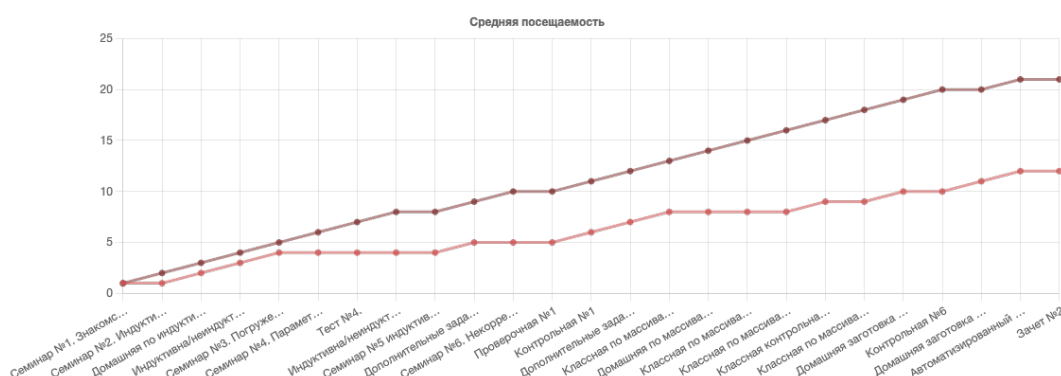


Рисунок 24 – Сравнение накопительной посещаемости отдельно двух студентов

Кроме того, в модуле предусмотрена возможность построения столбчатых диаграмм с упорядочиванием усредненных значений метрик в рамках выделенной сущности по убыванию или возрастанию. Такие графики позволяют оперативно выявлять студентов с наивысшими и наименьшими показателями, например, по среднему уровню плагиата за весь курс, что не отображается в модуле статистики результатов студентов, Рисунок 25. Для более удобного принятия решений по конкретным студентам их можно выделить отдельным цветом, что также может применяться для сравнения результатов студентов из двух разных групп.

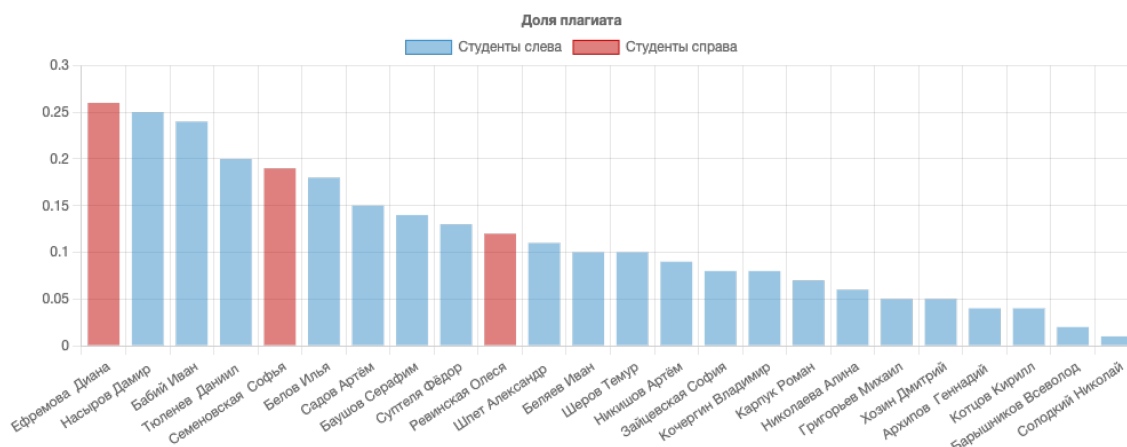


Рисунок 25 – Сравнение средней доли списанных задач студентов по курсу

Модуль также поддерживает сравнение групп, что особенно полезно для педагогических экспериментов. На графиках boxplot можно увидеть различия между группами в средних и медианных значениях, а также сделать выводы об общем разбросе результатов студентов, например, по метрикам доли баллов, Рисунок 26, и доли списанных задач, Рисунок 27. Если одна группа демонстрирует более высокие баллы при меньшем количестве списанных задач, это может указывать на более успешное освоение курса и являться подтверждением успешности проведения педагогического эксперимента.

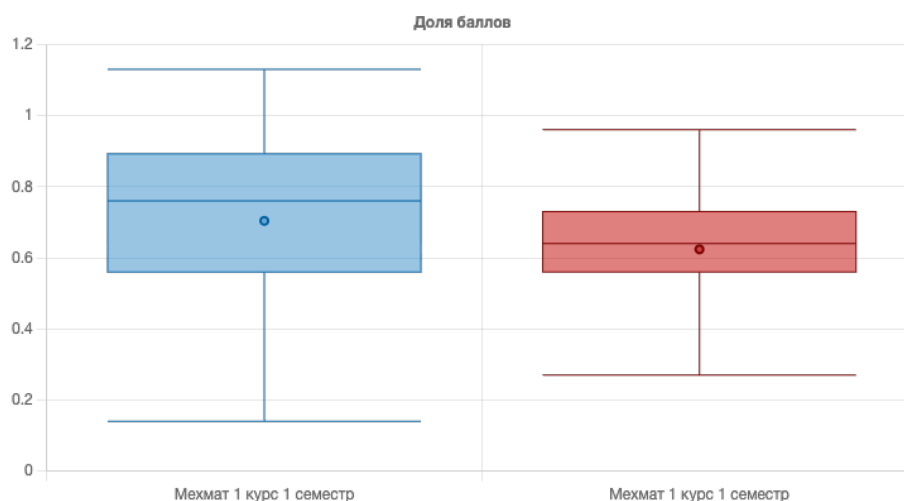


Рисунок 26 – Сравнение средней доли набранных баллов двух групп по курсу



Рисунок 27 – Сравнение средней доли списанных задач двух групп по курсу

Кроме того, модуль позволяет выдвигать гипотезы о структуре курса. Например, преподаватель может проверить, упорядочены ли контексты по сложности, анализируя график среднего времени решения задач. Если время решения увеличивается от контекста к контексту, это может свидетельствовать о возрастающей сложности заданий, Рисунок 28, что положительно сказывается на эффективности усвоения материалов студентами. В случае отсутствия такой зависимости преподаватель может принять решение о декомпозиции или упрощении отдельных блоков материала.

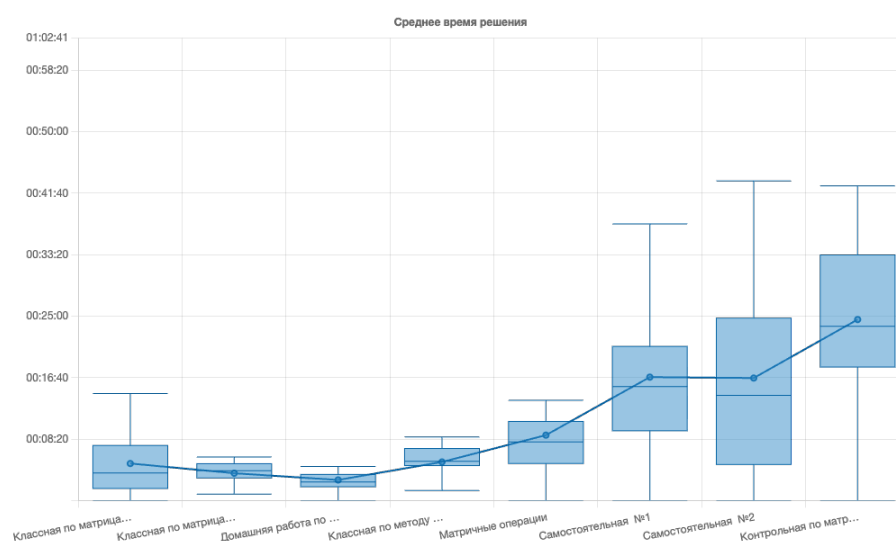


Рисунок 28 – Среднее время решения задач по контекстам в теме с возрастающей сложностью контекстов

3.3 Сигнатурные методы анализа временных рядов

Сигнатурные методы представляют собой эффективный инструмент для анализа временных рядов, который преобразует их в форму, удобную для задач машинного обучения. Сигнатурные методы являются одним из эффективных инструментов для обработки и выделения признаков из пути – функции, аргументом которой является время, например, траектории случайного процесса. С помощью этого инструмента можно преобразовать временные ряды в форму, более подходящую для задач машинного обучения, таких как классификация и прогнозирование.

Понятие сигнатуры пути впервые было введено в работах [54; 55] и представляет собой последовательность итерированных интегралов, которые отражают различные свойства пути. Известно, что при выполнении некоторых достаточно общих условий, наложенных на рассматриваемый класс путей, существует взаимно однозначное соответствие между сигнатурами и путями. В рамках различных практических направлений, от задач восстановления траекторий до анализа потоков данных различной природы, было установлено, что достижение приемлемого уровня точности восстановления пути по его сигнатуре зачастую возможно при использовании лишь начальных членов сигнатуры, а именно первых нескольких повторных интегралов. Таким образом, сигнатура временного ряда, как способ кодирования, оказывается инструментом высокоэффективной компрессии информации, заключённой в траектории.

Временные ряды, являющиеся представлением реальных данных, по сути представляются своими значениями в дискретном множестве моментов времени. Таким образом, все рассматриваемые траектории являются кусочно-линейными, что упрощает определение сигнатуры и использование ее свойств. В ранних работах рассматривались достаточно регулярные траектории – траектории с ограниченной вариацией, допускающие применение интегралов Римана-Стилтьеса. Для грубых траекторий (например, с показателем Гельдера меньше $1/2$), которые требуют применения стохастических интегралов или интегралов по

грубым траекториям, сигнатуры исследовались в работах [105; 104; 76]. Ряд результатов, например, взаимно-однозначное соответствие между траекториями и сигнатурами, остается верным и для грубых траекторий [153; 87].

На протяжении последнего десятилетия наблюдается стремительное проникновение сигнатурных методов в арсенал прикладного машинного обучения. Так, будучи скомбинированными с архитектурами сверточных нейронных сетей, они обеспечили получение первенства в престижном онлайн-соревновании ICDAR 2013 по распознаванию изолированных китайских иероглифов [83], а в сопряжении с моделью градиентного бустинга – заняли лидирующую позицию в состязании PhysioNet 2019, сосредоточенном на вычислительных аспектах кардиологической диагностики [152]. Кроме того, эти методы нашли широкое применение в области финансовой математики, в частности при решении задач хеджирования производных инструментов [56]. Для применения методов машинного обучения в условиях недостаточного количества эмпирических данных была предложена методология генерации дополнительных, статистически согласованных с оригиналом траекторий [38]. Сигнатурные методы успешно применяются в задачах классификации и регрессии. Например, в работе [86] сигнатуры используются для анализа финансовых временных рядов, где они позволяют выделять ключевые особенности данных. Авторы демонстрируют, что даже небольшое количество коэффициентов сигнатуры достаточно для классификации финансовых потоков и прогнозирования рыночных изменений. Традиционные методы анализа временных рядов, такие как динамические факторные модели (Dynamic Factor Models, DFM) и авторегрессионные модели (Autoregressive Integrated Moving Average, ARIMA), часто требуют строгих предположений о стационарности и линейности данных. В отличие от них, сигнатурные методы не накладывают таких ограничений. В работе [119] показано, что регрессия на сигнатурах превосходит DFM в задаче прогнозирования ВВП (валового внутреннего продукта), обеспечивая меньшую ошибку и большую устойчивость к нерегулярностям данных. Одной из проблем сигнатурных методов является экспоненциальный рост числа сигнатурных членов с увеличением уровня усечения. Для решения этой проблемы применяются методы

регуляризации, такие как LASSO и Elastic Net [165], а также методы уменьшения размерности, например, метод главных компонент (Principal Component Analysis, PCA). В работе [74] обсуждается, как выбор уровня усечения и стандартизация элементов сигнатуры влияют на качество моделей.

С теоретической точки зрения сигнатурный подход следует квалифицировать как непараметрическую, устойчивую к шумам методику извлечения репрезентативных признаков, способных впоследствии служить входными признаками для моделей машинного обучения. Его фундаментальная сила заключается в способности по параметризованному пути, задающему последовательность наблюдаемых состояний, сформировать компактный, но исчерпывающий набор признаков, аккумулирующих как аналитические, так и геометрические характеристики рассматриваемого процесса.

Определение и свойства пути.

Путь в R^d – это непрерывное отображение X из некоторого интервала $[a, b]$ в R^d . Чтобы подчеркнуть зависимость от времени используется обозначение $X_t = X(t): [a, b] \rightarrow R^d$.

В дальнейшем будем предполагать, что рассматриваемые пути являются достаточно «хорошими» отображениями, а именно они являются кусочно-дифференцируемыми (вообще говоря, справедливость большинства результатов сохранится, если считать, что пути имеют ограниченную вариацию). Будем называть путь гладким, если он бесконечно дифференцируем.

Интегралом от функции $f: R \mapsto R$ по одномерному пути $X: [a, b] \mapsto R$ называется следующая величина, формула (4):

$$\int_a^b f(X_t) dX_t = \int_a^b f(X_t) \dot{X}_t dt, \quad (4)$$

где последний интеграл является обычным (римановым) интегралом непрерывной ограниченной функции. Обозначение «верхняя точка» здесь и далее используется для дифференцирования по одной переменной: $\dot{X}_t = dX_t/dt$.

Заметим, что $f(X_t)$ тоже является путем на $[a, b]$, поэтому можно

естественным образом определить интеграл от пути по пути.

Интегралом от пути $Y: [a, b] \mapsto R$ по пути $X: [a, b] \mapsto R$ называется следующая величина, формула (5):

$$\int_a^b Y_t dX_t = \int_a^b Y_t \dot{X}_t dt. \quad (5)$$

3.3.1 Математический аппарат сигнатур путей

Обозначим координаты пути $X: [a, b] \mapsto R^d$ за $(X_t^1, \dots, X_t^d)$, где каждая координата $X^i: [a, b] \mapsto R$ является путем. Для каждого $i \in \{1, \dots, d\}$ и $t \in [a, b]$ определим следующую величину, формула (6)

$$S(X)_{a,t}^i = \int_{a < s < t} dX_s^i = X_t^i - X_a^i, \quad (6)$$

которая является приращением i -ой координаты пути до момента времени t . Отметим, что $S(X)_{a,\cdot}^i$ – это тоже путь.

Теперь определим для любой пары $i, j \in \{1, \dots, d\}$ двойной повторный интеграл, формула (7):

$$S(X)_{a,t}^{i,j} = \int_{a < s < t} S(X)_{a,s}^i dX_s^j = \int_{a < r < s < t} dX_r^i dX_s^j = \int_{a < s < t} \int_{a < r < s} dX_r^i dX_s^j. \quad (7)$$

Можно продолжить по индукции: для любого $k \geq 1$ и набора индексов $i_1, \dots, i_k \in \{1, \dots, d\}$ определим следующую величину, формула (8):

$$S(X)_{a,t}^{i_1, \dots, i_k} = \int_{a < s < t} S(X)_{a,s}^{i_1, \dots, i_{k-1}} dX_s^{i_k} = \int_{a < t_k < t} \dots \int_{a < t_1 < t_2} dX_{t_1}^{i_1} \dots dX_{t_k}^{i_k}. \quad (8)$$

Величина $S(X)_{a,t}^{i_1, \dots, i_k}$ называется k -кратным повторным интегралом от пути X по индексам $i_1, \dots, i_k$.

Сигнатурой пути $X: [a, b] \mapsto R^d$ называется бесконечный набор $S(X)_{a,b}$ всех повторных интегралов от X , формула (9):

$$S(X)_{a,b} = (1, S(X)_{a,b}^1, \dots, S(X)_{a,b}^d, S(X)_{a,b}^{1,1}, S(X)_{a,b}^{1,2}, \dots), \quad (9)$$

где первый элемент сигнатуры (соответствующий пустому индексу) по определению считается равным 1, а верхние индексы остальных элементов

пробегают набор всевозможных мульти-индексов, формула (10):

$$W = \{(i_1, \dots, i_k) \mid k \geq 1, i_1, \dots, i_k \in \{1, \dots, d\}\}. \quad (10)$$

Множество W называется множеством слов в алфавите $A = \{1, \dots, d\}$. Конечный набор чисел $S(X)_{a,b}^{i_1, \dots, i_k}$ для всевозможных мульти-индексов длины k будем называть k -ым уровнем сигнатуры.

Пример вычисления сигнатуры для одномерного пути. Рассмотрим произвольный одномерный путь $X: [a, b] \mapsto R$. Тогда сигнатура этого пути вычисляется следующим образом:

$$S(X)_{a,b}^1 = X_b - X_a,$$

$$S(X)_{a,b}^{1,1} = \frac{(X_b - X_a)^2}{2!},$$

$$S(X)_{a,b}^{1,1,1} = \frac{(X_b - X_a)^3}{3!},$$

...

$$S(X)_{a,b}^{1,1,\dots,1} = \frac{(X_b - X_a)^k}{k!}.$$

Отсюда можно видеть, что сигнатура по повторным индексам выражается через приращение по соответствующей координате, что верно и для многомерного случая.

Шафл-произведение.

Одной из глубоких алгебраических особенностей сигнатуры является то, что произведение двух её элементов $S(X)_{a,b}^{i_1, \dots, i_k}$ и $S(X)_{a,b}^{j_1, \dots, j_m}$ всегда может быть выражено через сумму других её элементов, которая зависит исключительно от мульти-индексов $(i_1, \dots, i_k)$ и $(j_1, \dots, j_m)$. Это свойство влечёт за собой важнейшее следствие, показывающее отсутствие алгебраической независимости между членами сигнатуры. Помимо этого, оно позволяет отказаться от непосредственного манипулирования произведениями элементов сигнатуры и перейти к работе с линейными комбинациями, что, в свою очередь, существенно упрощает их аналитическую обработку.

Далее вводится понятие шафл-произведения двух мульти-индексов.

Определение. Перестановка σ множества $\{1, \dots, k + m\}$ называется (k, m) -шафлом, если $\sigma^{-1}(1) < \dots < \sigma^{-1}(k)$ и $\sigma^{-1}(k + 1) < \dots < \sigma^{-1}(k + m)$. Для множества всех (k, m) -шафлов будем использовать обозначение $Shuffles(k, m)$.

Рассмотрим два мульти-индекса $I = (i_1, \dots, i_k)$ и $J = (j_1, \dots, j_m)$, $i_1, \dots, i_k, j_1, \dots, j_m \in \{1, \dots, d\}$. Определим мульти-индекс $(r_1, \dots, r_k, r_{k+1}, \dots, r_{k+m}) = (i_1, \dots, i_k, j_1, \dots, j_m)$.

Шафл-произведением I и J (обозначение: $I \# J$) называется конечный набор мульти-индексов длины $k+m$ следующего вида, формула (11):

$$I \# J = \{r_{\sigma(1)}, \dots, r_{\sigma(k+m)} \mid \sigma \in Shuffles(k, m)\}. \quad (11)$$

Теорема. Для любого пути $X: [a, b] \mapsto R^d$ и мульти-индексов $I = (i_1, \dots, i_k)$ и $J = (j_1, \dots, j_m)$, $i_1, \dots, i_k, j_1, \dots, j_m \in \{1, \dots, d\}$ верно равенство, формула (12):

$$S(X)_{a,b}^I S(X)_{a,b}^J = \sum_{K \in I \# J} S(X)_{a,b}^K. \quad (12)$$

Геометрический смысл сигнатур.

Независимость от начальной точки. Рассмотрим путь $X: [a, b] \mapsto R^d$ и $h \in R^d$. Пусть путь $Y: [x, y] \mapsto R^d$ имеет вид $Y_t = X_t + h$. Тогда $S(X)_{a,b} = S(Y)_{a,b}$.

Независимость от репараметризации времени. Рассмотрим путь $X: [a, b] \mapsto R^d$ и биективную непрерывную неубывающую функцию $\psi: [x, y] \mapsto [a, b]$. Пусть путь $Y: [x, y] \mapsto R^d$ имеет вид $Y_t = X_{\psi_t}$. Тогда $S(X)_{a,b} = S(Y)_{x,y}$.

Исходя из двух этих свойств, стоит подчеркнуть, что первый уровень сигнатуры $(S(X)_{a,b}^1, \dots, S(X)_{a,b}^d)$, как видно из её формального определения, совпадает с приращениями пути по отдельным координатам, в то время как второй уровень интерпретируется через понятие площади Леви – геометрической характеристики, указывающей на ориентированную площадь, охватываемую траекторией в двумерном пространстве. Пусть $X: [a, b] \mapsto R^2$ – двумерный путь, где $X_t = (X_t^1, X_t^2)$. Проведём прямую от начальной до конечной точки пути, после чего все получившиеся площади при пересечении этой прямой и пути рассмотрим со знаком минус, если они выше прямой, и со знаком плюс, если они ниже прямой.

Площадью Леви называется сумма данных площадей, Рисунок 29.

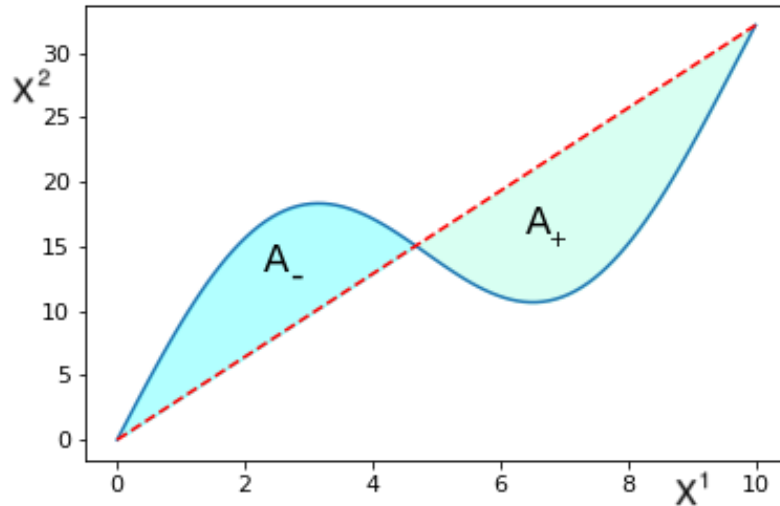


Рисунок 29 – Площадь Леви $A = A_+ - A_-$

Площадь Леви выражается через элементы сигнатуры следующим образом, формула (13):

$$A = \frac{1}{2} (S(X)_{a,b}^{1,2} - S(X)_{a,b}^{2,1}). \quad (13)$$

Тождество Чена.

Эффективность вычисления сигнатур в значительной мере зависит от природы исходного пути. В тех случаях, когда имеется аналитически заданная параметризация, соответствующие повторные интегралы могут быть получены в явной форме. Однако в большинстве прикладных задач, где траектории представляют собой последовательности наблюдений, их форма носит кусочно-линейный характер, а параметризация затруднена или вовсе отсутствует. В подобных случаях оказывается целесообразным применение тождества Чена, позволяющего рекурсивно вычислять сигнатуру всего пути по сигнатурам его составных отрезков.

Теорема (тождество Чена). Пусть $a < b < c$ и $X: [a, c] \mapsto R^d$. Тогда для любых $(i_1, \dots, i_k) \in W$ выполнено равенство, формула (14):

$$S(X)_{a,c}^{i_1, \dots, i_k} = \sum_{m=0}^k S(X)_{a,b}^{i_1, \dots, i_m} S(X)_{b,c}^{i_{m+1}, \dots, i_k}. \quad (14)$$

Для каждого одномерного сегмента сигнатура может быть получена напрямую, как показано в примере вычисления сигнатуры одномерного пути, после чего производится поочерёдное соединение с предыдущими с использованием вышеуказанного тождества.

3.3.2 Логарифмические сигнатуры и их свойства

Из теоремы о шафл-произведении вытекает, что элементы сигнатуры не являются алгебраически независимыми, однако для многих моделей машинного обучения требуются независимые входные данные для корректной работы и лучшего качества. Эту проблему решает понятие логарифмических сигнатур, которые, в отличие от обычных сигнатур, представляют собой алгебраически независимый набор величин, подходящих для прямого использования в качестве признакового пространства моделей.

Для определения понятия логарифмической сигнатуры введем сначала алгебру формальных степенных рядов.

Рассмотрим d формальных неопределённых величин $e_1, \dots, e_d$. Алгеброй некоммутативных формальных степенных рядов из d неопределённых называется векторное пространство всех рядов следующего вида, формула (15):

$$\sum_{k=0}^{\infty} \sum_{i_1, \dots, i_k=1}^d \lambda_{i_1, \dots, i_k} e_{i_1} \dots e_{i_k}, \quad (15)$$

где параметр второй суммы пробегает по всем мульти-индексам $(i_1, \dots, i_k), i_1, \dots, i_k \in 1, \dots, d$, и коэффициенты $\lambda_{i_1, \dots, i_k}$ являются вещественными числами.

На этом пространстве определены стандартные операции сложения рядов и умножения ряда на коэффициент, а также операция умножения рядов $\otimes$. При этом элементы пространства некоммутативны, т.е., например, $e_1 e_2$ и $e_2 e_1$ являются разными элементами.

Сигнатура может быть «закодирована» как элемент этого пространства (т.е. сигнатура однозначно записывается в виде формального степенного ряда), формула (16):

$$S(X)_{a,b} = \sum_{k=0}^{\infty} \sum_{i_1, \dots, i_k=1}^d S(X)_{a,b}^{i_1, \dots, i_k} e_{i_1} \dots e_{i_k}. \quad (16)$$

Иными словами, элементы сигнатуры рассматриваются как соответствующие коэффициенты ряда, что позволяет «перейти» в алгебру рядов. В ней можно выполнять операции сложения и умножения рядов, а коэффициенты полученного ряда интерпретировать как элементы сигнатуры.

Далее для формального ряда, в котором первый коэффициент равен 1, а остальные равны 0 (что соответствует сигнатуре $\{1, 0, 0, \dots\}$) будем использовать обозначение 1.

Логарифмической сигнатурой пути $X: [a, b] \mapsto R^d$ называется следующий формальный степенной ряд, формула (17)

$$\log S(X)_{a,b} = - \sum_{n=1}^{\infty} \frac{1}{n} (1 - S(X)_{a,b})^{\otimes n}. \quad (17)$$

Тем самым, логарифмическая сигнатура становится мощным инструментом генерации независимого набора признаков, обладающих полной описательной способностью по отношению к геометрии исходного многомерного пути.

Следует, однако, указать, что сигнатура не позволяет восстановить траекторию в полном объёме. Так, например, информация о скорости прохождения пути утрачивается в силу инвариантности сигнатуры относительно репараметризации времени. Тем не менее, при соблюдении определённых условий, в частности, при отсутствии самопересечений, сигнатура полностью определяет как множество точек, через которые проходит путь (с точностью до параллельного переноса), так и порядок их обхода, что открывает возможности для применения в задачах восстановления структуры данных.

3.4 Генерация синтетических данных, сохраняющих статистические свойства исходных траекторий

Практика предоставляет множество различных задач, в которых неизвестно распределение и стохастическая основа данных, в связи с чем нет возможности просчитать математические свойства модели. Описывающие эти задачи модели можно значительно усложнять, подбирая и добавляя в них новые параметры, однако в реальности они не могут полностью описать наблюдаемое поведение.

Помимо этого, при использовании временных рядов во многих ситуациях может оказаться недостаточное количество данных для применения моделей глубокого машинного обучения, а также данные могут быть конфиденциальны и недоступны для обучения моделей. Примерами этого могут являться образовательные, медицинские и финансовые данные.

В ситуациях, когда необходимо расширить доступное множество данных, не прибегая к гипотезам об их вероятностной природе, предлагается использовать сигнатурные методы в сочетании с генеративными моделями. Так, на основе оригинального множества временных рядов, представленных в сигнатурной форме, можно синтезировать новые данные, демонстрирующие схожее распределение. При этом необходимость знания истинного распределения отпадает, поскольку оценка схожести может быть проведена с помощью, например, метрики максимального среднего расхождения (MMD), построенной также на основе сигнатур [57].

Данные образовательного цифрового следа студентов являются ярким примером временных рядов, у которых неизвестно их распределение и стохастическая основа. Помимо этого, в зависимости от университета, курса и среднего уровня студентов они могут отличаться, что вводит еще большие ограничения на количество доступных данных. Поэтому для тестирования различных педагогических гипотез, а также для создания моделей по подбору персональных траекторий и построению поведенческой и предсказательной аналитики, необходимо много лет собирать примеры студенческих данных.

В связи с этим были применены методы генерации, анализа и прогноза траекторий обучения студентов, использующие сигнатуры исходного набора данных. Преимуществом использования сигнатурных методов является то, что они способны улавливать взаимосвязи между различными размерностями временного ряда. Например, появляется возможность обнаруживать взаимные зависимости между посещаемостью, временем, потраченным на решение задач, плагиатом, долей набранных за занятия баллов и другими метриками цифрового следа студентов. Это значительно повышает общее качество модели, поскольку рассматривание отдельных характеристик независимо друг от друга не дает полной картины. Например, у студента могут быть высокие баллы, но при высоком уровне плагиата, что меняет его общий статус относительно группы. Помимо этого, у модели появляется возможность улавливать, например, повышение плагиата при падении времени решения задач, падение среднего балла при падении посещаемости, и другие зависимости.

Такой подход, реализованный в контексте образовательной аналитики, позволяет, используя массив синтетически сгенерированных с помощью сигнатурных методов данных, построить модель прогнозирования поведения студента, в которой сигнатуры временных рядов выступают в роли признакового пространства. Эта модель, в свою очередь, служит инструментом визуализации и интерпретации динамики учебных результатов, указывая преподавателю на необходимость индивидуального вмешательства или корректировки образовательной траектории с целью максимизации обучающего эффекта. Более того, на основе такого прогностического механизма можно автоматически формировать персонализированные образовательные траектории, предлагая учащимся релевантные задания, а также вспомогательные материалы, соответствующие их текущему уровню и способствующие увеличению эффективности их обучения.

Таким образом, на основе исходных данных цифрового следа обучающихся формируется набор логарифмических сигнатур соответствующих временных рядов. Эти сигнатуры поступают на вход вариационного автокодировщика с целью

генерации дополнительного множества синтетических логарифмических сигнатур [38]. Полученное расширенное множество может быть непосредственно использовано, в частности, для обучения или тестирования других моделей. Если же требуется восстановить сами совместные траектории, к сгенерированным логарифмическим сигнатурам применяется алгоритм обратного преобразования.

3.4.1 Обращение логарифмической сигнатуры, теорема об оценке погрешности

Для обращения логарифмической сигнатуры предлагается использовать эволюционный алгоритм подбора нужного пути, чтобы его логарифмическая сигнатура была наиболее похожа на изначальную. Если распределения приращений каждой координаты пути известны, их можно передать в модель в качестве параметров. Однако такое решение подойдет только для теоретической модели, ведь на практике распределения приращений траекторий чаще всего неизвестны. Пусть мы ищем путь $X: [a, b] \mapsto R^d$, $(X_t^1, \dots, X_t^d)$, где каждая координата $X^i: [a, b] \mapsto R$. В общем случае путь не является кусочно-линейным и может принимать любые вещественные значения. Требуется найти кусочно-линейный путь $\hat{X}: [a, b] \mapsto R^d$, который наилучшим образом приближает путь X в смысле минимизации метрики, формула (18):

$$\sup_{t \in [a, b]} |X(t) - \hat{X}(t)| = \sup_{t \in [a, b]} \left(\sum_{i=1}^d |X_i(t) - \hat{X}_i(t)| \right). \quad (18)$$

Известно, что любой непрерывный путь можно аппроксимировать сколь угодно близко кусочно-линейным путем, однако в рамках данной модели у нас есть ряд ограничений. Координаты "изломов" $\hat{X}$ можно рассматривать только с фиксированным шагом Δt , который отвечает за то, с каким промежутком времени делаются измерения метрик. Помимо этого, на практике возможно подбирать значения приращения каждой из координат пути $\hat{X}_i$ только из ограниченной сетки $\{-n_i \cdot h_i, \dots, n_i \cdot h_i\}$ с шагом h_i . Если взять слишком маленькие h_i и слишком большие n_i – то алгоритм будет сходиться слишком долго. Поэтому в зависимости

от реальных целей необходимо ограничить значения h_i и n_i для каждой координаты пути. При этом при ограничении n_i накладывается ограничение на модуль производной приближаемого пути в зависимости от значения Δt : чем меньше Δt , тем пути с большей производной возможно приблизить при фиксированных h_i и n_i . Итоговое утверждение можно сформулировать следующим образом.

Теорема об оценке погрешности.

Пусть даны фиксированные числа $h_i \in R_+, n_i \in N, \Delta t \in R_+$.

Пусть $X: [a, b] \mapsto R^d$ – непрерывный и кусочно-дифференцируемый путь, у которого $\forall i: |X'_i(t)| \leq \frac{h_i n_i}{\Delta t}$. Тогда существует кусочно-линейный путь $\hat{X}: [a, b] \mapsto R^d$ с приращениями по каждой координате i со значениями из множества $\{-n_i \cdot h_i, \dots, 0, \dots, n_i \cdot h_i\}$ и "изломами" в точках $k \cdot \Delta t, k \in Z$ такой, что:

$$\forall i: \sup_{t \in [a, b]} |X_i(t) - \hat{X}_i(t)| \leq h_i \cdot \left(2n_i + \frac{1}{2}\right).$$

Доказательство. Построим у искомого пути $\hat{X}_i$ значения в узловых точках с координатами $k \cdot \Delta t$ следующим образом: возьмем значения пути X_i в этих точках и выберем к ним ближайшие точки со значениями из сетки $\{h_i \cdot l, l \in Z\}$.

Таким образом, из построения $\hat{X}_i$ получаем, что: $|X_i(k \cdot \Delta t) - \hat{X}_i(k \cdot \Delta t)| \leq \frac{h_i}{2}$, так как в узловых точках значение пути $X_i(k \cdot \Delta t)$ может лежать только в окрестности $\left[-\frac{h_i}{2}, \frac{h_i}{2}\right]$ значения пути $\hat{X}_i(k \cdot \Delta t)$ (иначе в этой точке можно было бы увеличить или уменьшить значение подбираемого пути $\hat{X}_i$ так, чтобы значение исходного пути попало бы в эту окрестность его значения).

На каждом из отрезков $[k \cdot \Delta t, (k+1) \cdot \Delta t]$ по теореме Лагранжа для каждой координаты пути X_i имеем:

$$X_i(t) = X_i(k \cdot \Delta t) + \xi \cdot (t - k \cdot \Delta t),$$

$$\hat{X}_i(t) = \hat{X}_i(k \cdot \Delta t) + \hat{\xi} \cdot (t - k \cdot \Delta t),$$

где ξ и $\hat{\xi}$ – значения производных $X_i(t)$ и $\hat{X}_i(t)$ соответственно в некоторых точках на отрезке $[k \cdot \Delta t, t]$.

Имеем $|\xi| \leq \frac{h_i \cdot n_i}{\Delta t}$ по формулировке предложения.

Также $|\hat{\xi}| \leq \frac{h_i \cdot n_i}{\Delta t}$, потому что приращения $\hat{X}_l(t)$ на отрезках длиной Δt берутся из множества $\{-n_i \cdot h_i, \dots, n_i \cdot h_i\}$, поэтому неравенство следует из определения производной как тангенса угла наклона касательной.

Из построения $\hat{X}_l$ имеем, что $|X_i(k \cdot \Delta t) - \hat{X}_l(k \cdot \Delta t)| \leq \frac{h_i}{2}$.

Итого, получаем:

$$\begin{aligned} |X_i(t) - \hat{X}_l(t)| &\leq |X_i(k \cdot \Delta t) - \hat{X}_l(k \cdot \Delta t)| + |t - k \cdot \Delta t| |\xi - \hat{\xi}| \leq \\ &\leq \frac{h_i}{2} + \Delta t \cdot 2 \frac{h_i n_i}{\Delta t} = h_i \cdot \left(2n_i + \frac{1}{2}\right). \end{aligned}$$

Замечание 1.

Если провести достаточное количество итераций генерирования приращений каждой координаты пути $\hat{X}_l$ из арифметического распределения со значениями из соответствующего множества $\{-n_i \cdot h_i, \dots, n_i \cdot h_i\}$, то с вероятностью, равной единице, в один момент будет получен оптимальный путь $\hat{X}$.

Замечание 2.

Итоговую метрику разности исходного пути и его приближения можно оценить следующим образом, формула (19):

$$\sup_{t \in [a, b]} |X(t) - \hat{X}(t)| = \sup_{t \in [a, b]} \left(\sum_{i=1}^d |X_i(t) - \hat{X}_l(t)| \right) \leq d \cdot \max_i \left(h_i \cdot \left(2n_i + \frac{1}{2}\right) \right). \quad (19)$$

3.4.2 Эволюционный алгоритм обращения логарифмической сигнатуры

Перейдём к практическому применению данной теории. Необходимо оптимальным образом подобрать параметры h_i и n_i для каждой координаты пути. Параметр h_i , отвечающий за окрестность, в пределах которой отождествляются значения приращений, подбирается исходя из реально поставленной задачи – в пределах какой погрешности изменения метрики не имеют значения.

Параметр n_i , отвечающий за то, какой максимальный скачок может быть сделан за один шаг времени (ведь $|h_i \cdot n_i|$ – это модуль максимально возможного

приращения, при этом h_i уже фиксировано), подбирается тоже исходя из реально поставленной задачи — насколько сильно может быть ограничен класс генерируемых путей. Если выбрать значение параметра слишком большим, то подбор будет идти путями со слишком большим размахом приращений, что затрудняет сходимость алгоритма, а если сделать его слишком маленьким, то теряется возможность приблизить пути с большими приращениями. Для помощи в подборе этого параметра предлагается посчитать среднее и максимальное приращение по каждой из координат. После того, как подобраны нужные распределения и их параметры, подбираются оптимальным образом количество путей в популяции и количество итераций алгоритма эволюции. Эти параметры необходимо подбирать в зависимости от допустимых ограничений по времени работы алгоритма: чем они больше, тем дольше будет выполняться обращение одной логарифмической сигнатуры. Далее у каждого пути из текущей популяции считается логарифмическая сигнатура и смотрится метрика loss разницы исходной логарифмической сигнатуры и посчитанной, по этой метрике пути сортируются и выбирается настраиваемый процент лучших.

Теперь лучшие пути нужно размножить до исходного размера популяции. Для этого, пока не получено изначальное количество путей, из лучших случайно выбираются два различных пути, после чего вызывается оператор их сложения, который заключается в том, что с вероятностью 0.5 берутся приращения одного или другого пути. После этого дополнительно вызывается функция "мутации" полученного пути. В процессе этой функции настраиваемый процент приращений заново определяется случайным образом из заданных в модели распределений. После этого считается путь, как последовательная сумма его приращений, и к нему применяется заданное в модели преобразование временного ряда. Этот процесс повторяется с новой популяцией заданное количество итераций. В итоге модель возвращает путь с логарифмической сигнатурой, наиболее близкой к исходной.

Далее приведен псевдокод эволюционного алгоритма обращения логарифмической сигнатуры.

Require: s^* – целевая логарифмическая сигнатура;

N – размер популяции;

T – число итераций;

$p \in (0,1)$ – доля отбора (top- p);

$\mu \in (0,1)$ – вероятность мутации;

L – число временных шагов;

d – размерность пространства;

$\mathcal{D}_i$ – распределения приращений по координатам;

k – порядок логарифмической сигнатуры

Ensure: $\tilde{X}^*$ – путь, минимизирующий расхождение логарифмической сигнатуры.

// Инициализация популяции

for $j = 1, \dots, N$ **do**

for $t = 1, \dots, L$ **do**

 Сгенерировать приращение: $\Delta X_t^{(j)} = (\xi_1, \dots, \xi_d)$, $\xi_i \sim \mathcal{D}_i$

end for

 Построить путь как накопленную сумму: $\tilde{X}^{(j)} \leftarrow cumsum(\Delta X^{(j)})$

end for

// Основной цикл эволюции

for $\tau = 1, \dots, T$ **do**

for $j = 1, \dots, N$ **do**

 Вычислить логарифмическую сигнатуру: $s^{(j)} \leftarrow LogSig_k(\tilde{X}^{(j)})$

 Вычислить функцию потерь:

$$l^{(j)} \leftarrow \frac{1}{m} \sum_{r=1}^m \frac{|s_r^{(j)} - s_r^*|}{|s_r^*| \cdot (1 + r)}$$

end for

Отсортировать популяцию по возрастанию $l^{(j)}$

$\mathcal{P}$ первые $\lfloor pN \rfloor$ особей

$\mathcal{P}_{new} \leftarrow \mathcal{P}$

while $|\mathcal{P}_{new}| < N$ **do**

 Случайно выбрать двух различных родителей: $X^{(a)}, X^{(b)} \in \mathcal{P}, a \neq b$

for $t = 1, \dots, L$ **do**

$$\Delta X_t^{(ch)} \leftarrow \begin{cases} \Delta X_t^{(a)}, & \text{с вероятностью 0.5} \\ \Delta X_t^{(b)}, & \text{иначе} \end{cases}$$

end for

for $t = 1, \dots, L$ **do**

if $Bernoulli(\mu) = 1$ **then**

 Перегенерировать приращение:

$$\Delta X_t^{(ch)} \leftarrow (\xi_1, \dots, \xi_d), \quad \xi_i \sim \mathcal{D}_i$$

end if

end for

```

        Построить путь потомка:  $\tilde{X}^{(ch)} \leftarrow \text{cumsum}(\Delta X^{(ch)})$ 
        Добавить потомка в популяцию:  $\mathcal{P}_{new} \leftarrow \mathcal{P}_{new} \cup \{\tilde{X}^{(ch)}\}$ 
    end while
    Обновить популяцию:  $\mathcal{P} \leftarrow \mathcal{P}_{new}$ 
end for
// Возврат результата
return  $\tilde{X}^* \leftarrow \arg \min_j l^{(j)}$ 

```

Выводы по главе 3

В третьей главе проведено исследование и разработаны математические методы и алгоритмы анализа цифрового следа обучающихся, направленные на мониторинг, интерпретацию и прогнозирование образовательных траекторий. В ходе исследования показано, что цифровой след обучающихся, формируемый в процессе взаимодействия с элементами образовательной платформы, представляет собой многомерный и неравномерный во времени массив данных, требующий использования специализированных методов анализа.

Сформулирован подход к сбору, хранению и визуализации аналитических данных, обеспечивающий целостное представление цифрового следа студентов и его доступность для последующего анализа. Разработан модуль аналитики цифрового следа, который позволяет интегрировать данные из различных компонентов образовательной платформы, обеспечивать их согласованность и использовать как для оперативного мониторинга, так и для углублённого анализа учебной деятельности.

Рассмотрено применение сигнатурных методов анализа временных рядов, позволяющих формализовать траектории образовательной активности обучающихся. Изучен математический аппарат сигнатур путей, показаны их свойства и преимущества при работе с последовательностями событий различной длины и структуры. Установлено, что сигнатурное представление позволяет компактно и информативно кодировать динамику учебной активности обучающихся, сохраняя информацию о порядке и взаимосвязи событий.

В результате исследования задачи генерации синтетических данных, сохраняющих статистические и структурные свойства исходных образовательных

траекторий, были рассмотрены и разработаны методы обращения логарифмической сигнатуры, а также доказана теорема об оценке погрешности, возникающей при восстановлении исходных путей по их сигнатурам. Предложен эволюционный алгоритм обращения логарифмической сигнатуры, позволяющий устойчиво приближать исходные временные ряды и формировать синтетические данные, пригодные для тестирования и валидации аналитических моделей. Показано, что применение синтетических траекторий с характеристиками, близкими к реальным данным, позволяет расширять обучающие выборки и уменьшать зависимость аналитических моделей от ограниченного объёма наблюдений. Это особенно актуально для образовательных систем, где доступ к полным и репрезентативным данным нередко ограничивается нормативными и организационными требованиями.

Показано, что предложенные математические методы и алгоритмы обеспечивают переход к более глубокому и формализованному анализу цифрового следа обучающихся по сравнению с традиционными подходами, основанными на агрегированной статистике. Разработанные решения позволяют не только описывать текущие состояния образовательного процесса, но и осуществлять прогнозирование и поддержку адаптивного управления обучением.

ГЛАВА 4. ИНТЕГРАЦИЯ И АПРОБАЦИЯ РЕАЛИЗОВАННЫХ МОДУЛЕЙ НА БАЗЕ ЦИФРОВОЙ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ МИРЕРА

В предыдущих главах были обоснованы структура и состав универсальных модулей автоматизации образовательного процесса, а также разработаны алгоритмы их функционирования. Предложенные решения обладают свойством платформенной независимости и могут быть интегрированы в различные цифровые образовательные системы. Вместе с тем, для подтверждения их практической применимости необходима экспериментальная проверка в условиях реальной образовательной платформы.

В качестве экспериментальной базы для апробации предложенных решений и проверки их универсальности была выбрана цифровая образовательная платформа Мирера. Архитектура платформы соответствует типовой структуре и включает в себя модули управления учебным контентом, взаимодействия пользователей и сбора цифрового следа. Система предусматривает возможность расширения функциональности за счёт подключения дополнительных модулей, что особенно важно при экспериментальном внедрении и развитии платформы. Цифровая образовательная платформа Мирера ориентирована на поддержку преподавателя и автоматизацию рутинных процессов, в том числе с использованием технологий искусственного интеллекта, при сохранении ключевой роли преподавателя в образовательном процессе. Платформа поддерживает проведение интерактивных занятий, содержащих большое количество заданий с автоматической проверкой по различным дисциплинам, включая программирование, математику, а также финансовые, экономические и социально-гуманитарные направления. Совместное использование этих инструментов и организация эффективного взаимодействия между преподавателем и обучающимися способствуют повышению эффективности освоения учебного материала. При любом формате обучения собирается детальная поведенческая аналитика и цифровой след, используемые для корректировки учебного процесса и построения адаптивных траекторий.

4.1 Архитектура платформы и механизмы интеграции разработанных интеллектуальных модулей

Проектирование цифровой образовательной платформы Мирера основано на принципах модульности, расширяемости и независимого развития функциональных компонентов. Такой подход позволяет адаптировать платформу к различным образовательным сценариям, обеспечивать масштабирование системы, а также внедрять новые интеллектуальные модули без необходимости переработки базовой инфраструктуры. Архитектура платформы ориентирована на построение единого образовательного пространства, в рамках которого возможно не только развитие собственных компонентов, но и подключение внешних специализированных цифровых образовательных сред без дублирования их функциональности. Архитектура платформы и интеграции интеллектуальных модулей и внешних систем представлены на рисунке, Рисунок 30.

Технологическая реализация платформы опирается на современные веб-технологии. Основным языком разработки – JavaScript. Клиентская часть реализована с использованием фреймворка Vue, обеспечивающего реактивность пользовательского интерфейса и динамическое обновление данных. Серверная часть функционирует в среде Node.js с использованием фреймворка Express, реализующего API для взаимодействия между компонентами системы. Хранение данных организовано с использованием нереляционной базы данных MongoDB, что обеспечивает гибкость при работе с неструктурированными и слабо структурированными данными. Для хранения файлов большого объёма, включая мультимедийные данные, применяется объектное хранилище MinIO, совместимое с S3-интерфейсом. Взаимодействие между компонентами системы и реализация асинхронной обработки данных обеспечивается посредством брокера сообщений RabbitMQ, что позволяет реализовать событийно-ориентированную архитектуру и повысить масштабируемость системы. Отдельные модули, в частности связанные с обработкой данных и применением методов машинного обучения, реализованы на языке Python.

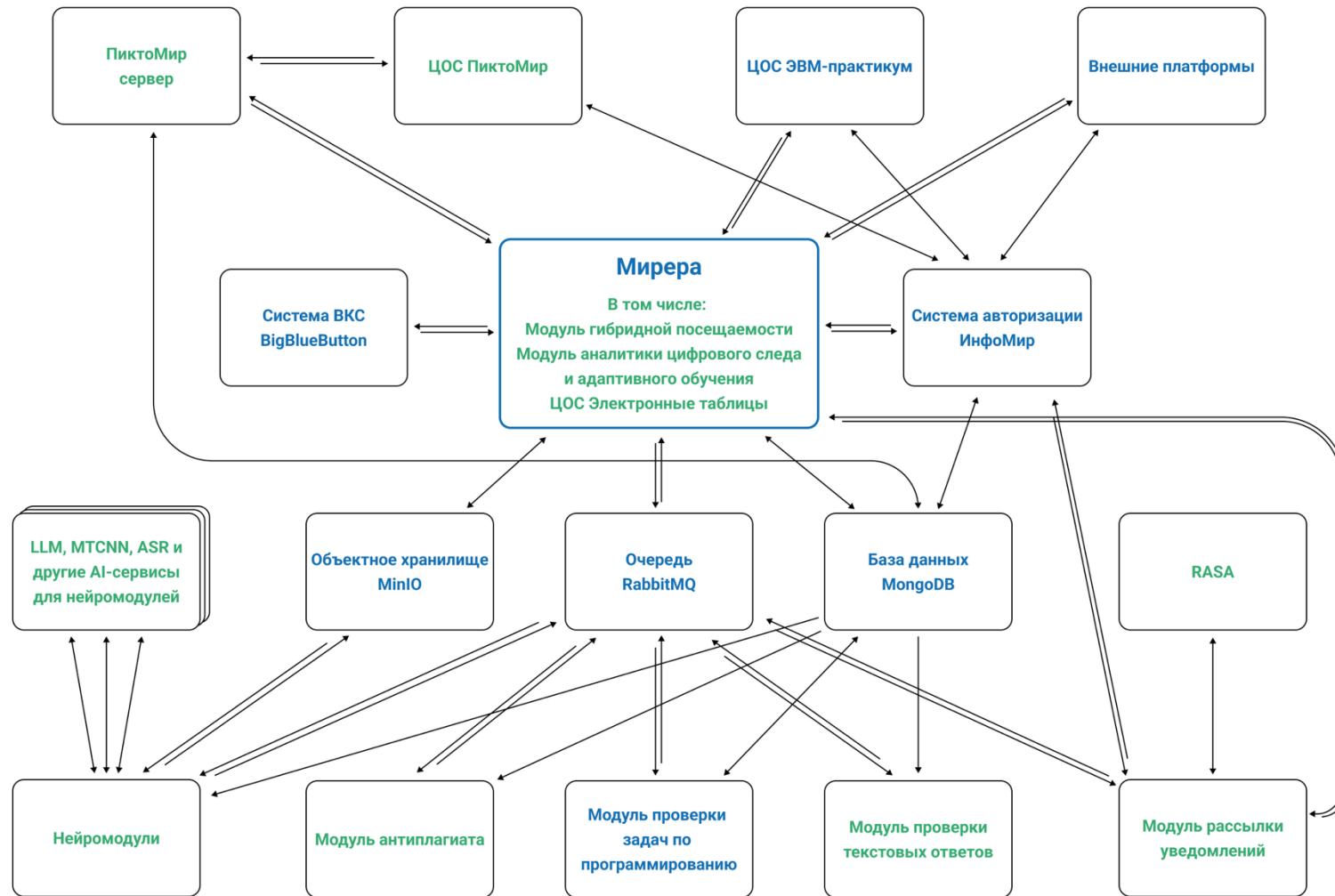


Рисунок 30 – Архитектура цифровой образовательной платформы Мирера. Разработанные и описанные в рамках диссертационной работы модули выделены зеленым цветом

Анализ архитектуры показывает, что платформа представляет собой распределённую модульную систему. Взаимодействие компонентов в ней осуществляется через формализованные интерфейсы и событийные механизмы, а центральным элементом выступает ядро платформы, которое координирует модули, обрабатывает данные и реализует логику. Такая организация обеспечивает изоляцию функциональных компонентов, их независимое развитие и возможность масштабирования без нарушения целостности системы. В рамках проведённой апробации разработанные модули интегрировались в платформу с использованием различных механизмов, определяемых их функциональными особенностями и требованиями к обработке данных. При этом сама архитектура не накладывает жёстких ограничений на способ подключения компонентов, что позволяет реализовывать интеграцию как на уровне данных и сервисов, так и на уровне событийного взаимодействия. Следует отметить, что выявленные в процессе внедрения особенности интеграции носят не частный характер, связанный исключительно с реализацией платформы Мирера, а отражают более общие закономерности построения современных цифровых образовательных систем. В частности, речь идёт о необходимости обеспечения совместимости разнородных компонентов, согласования форматов данных, а также организации взаимодействия между модулями платформы и внешними цифровыми средами.

Интеграция различных модулей ориентирована на включение относительно изолированных компонентов в общую логику обработки данных и управления образовательным процессом. В частности, интеграция внутренних модулей, реализуемых в рамках платформы, осуществляется с использованием единой инфраструктуры обработки данных и событий, что обеспечивает их согласованное взаимодействие и совместимость. Примерами полностью внутренней интеграции являются модуль учета посещаемости и модуль аналитики цифрового следа обучающихся. Данные модули реализуются в рамках единой архитектуры платформы и используют общую инфраструктуру хранения и обработки данных. Их функционирование основано на обработке событий, формируемых внутри системы, таких как действия пользователей, результаты выполнения заданий и

данные об участии в образовательных активностях.

В то же время интеграция внешних сервисов и интеллектуальных модулей требует использования специализированных механизмов взаимодействия. Взаимодействие с такими компонентами, как правило, осуществляется через API и механизмы обмена сообщениями, что позволяет обеспечить их включение в архитектуру системы без нарушения её целостности. В качестве примеров внешней интеграции можно рассмотреть ряд специализированных модулей, реализованных как автономные сервисы. К ним относятся модуль антиплагиата, модуль проверки текстовых ответов, нейромодули и модуль рассылки уведомлений, взаимодействующие с платформой через очередь сообщений и API.

Модуль антиплагиата реализован как специализированный сервис асинхронной обработки, интегрированный в событийно-ориентированную архитектуру платформы. Входной точкой для него служит очередь сообщений в RabbitMQ, из которой извлекается информация о необходимости проверки конкретного решения. После получения сообщения модуль обращается к базе данных платформы для извлечения совокупности решений по соответствующей задаче. На основе полученных данных выполняется процедура сравнительного анализа, направленная на выявление признаков заимствования. Результаты проверки формируются в структурированном виде и публикуются обратно в очередь ответов, откуда основной сервис системы осуществляет их последующую обработку, включая сохранение в базе данных и отображение пользователю.

Модуль проверки текстовых ответов функционирует по аналогичной модели и использует идентичный механизм взаимодействия с другими компонентами системы. Он также получает задания через очередь сообщений, извлекает необходимые данные из базы и выполняет их анализ в соответствии с заданными критериями. Итоговые результаты передаются обратно через очередь, обеспечивая единообразие интеграционного подхода.

Роутер нейромодулей выступает в роли координирующего компонента, реализующего сложные сценарии обработки данных с использованием нейросетевых сервисов. Получая задания через очередь сообщений, он

декомпозирует исходную задачу на подзадачи и отправляет их в соответствующие очереди, а когда уже подзадача попадает к нему, он агрегирует необходимые входные данные, извлекая их как из базы данных, так и из объектного хранилища, и направляет их в соответствующий специализированный сервис с подходящей нейросетевой моделью. По мере получения промежуточных результатов нейрорouter принимает решения о дальнейшем ходе выполнения: либо инициирует последующие этапы обработки в рамках текущего пайплайна, либо формирует итоговый результат и передает его в основной сервис системы. В случаях, когда результат обработки представляет собой крупные бинарные данные (например, видеоматериалы), он сохраняется в объектном хранилище, что позволяет оптимизировать использование вычислительных и сетевых ресурсов.

Модуль рассылки уведомлений представляет собой сервис, ориентированный на обработку событий, связанных с необходимостью информирования пользователей. Он получает сообщения из очереди, содержащие информацию о требуемых уведомлениях, после чего инициирует серию запросов к API основной системы для получения данных о соответствующих сущностях. Дополнительно осуществляется взаимодействие с системой авторизации с целью определения предпочтительного канала доставки уведомлений для каждого пользователя. Отправка уведомлений реализуется с учетом этих предпочтений. Кроме того, модуль использует API для принятия заявок, например, на участие в учебных группах или на преподавание. Помимо этого, модуль обеспечивает обработку входящих запросов от пользовательских ботов, взаимодействующих с диалоговой системой на базе Rasa по протоколу HTTP.

4.2 Архитектурные подходы к интеграции цифровых образовательных сред

В условиях интенсивной цифровизации образовательного процесса требуется не только внедрение отдельных электронных ресурсов, но и построение целостных инфраструктур, объединяющих разнообразные учебные среды в единое пространство. Самостоятельные цифровые инструменты, такие как визуальные редакторы, симуляторы, языковые среды и интеллектуальные тестирующие

комплексы, при отсутствии взаимной интеграции оказываются функционально изолированными. Разработка интеллектуальных образовательных платформ требует включения в их состав различных предметных цифровых сред, каждая из которых предоставляет специфические средства для освоения той или иной дисциплины. Однако простое сосуществование этих сред в рамках одного образовательного курса не даёт желаемого эффекта – необходимы механизмы интеграции, которые позволяли бы синхронизировать данные, унифицировать аутентификацию пользователей, а также автоматизировать проверку решений и ведение статистики.

Цифровая образовательная платформа Мирера обеспечивает согласованную работу с разнообразными цифровыми образовательными средами. В архитектуре платформы предусмотрено несколько уровней взаимодействия, каждый из которых соотнесён с функциональными особенностями встраиваемой среды. Реализованы четыре концептуально различающиеся модели интеграции, каждая из которых соответствует определённой роли подключаемой цифровой среды и характеру её взаимодействия с платформой. Первая модель предполагает внешнюю интеграцию с инициацией со стороны платформы. В этом случае цифровая среда работает автономно, сохраняя собственный интерфейс и внутреннюю логику, тогда как платформа отвечает за создание и синхронизацию заданий, результатов и аутентификацию пользователей. Вторая модель основана на внешне-инициированной интеграции, при которой подключаемая цифровая среда самостоятельно управляет образовательным контентом, формирует структуру заданий и передаёт данные в платформу. Платформа, в свою очередь, выполняет функции отображения, контроля и сбора статистики. Третья модель представляет собой внутреннюю интеграцию, при которой программное ядро среды встраивается непосредственно в платформу и становится частью её интерфейса и инфраструктуры. Это обеспечивает единообразный пользовательский опыт и централизованную проверку решений. Дополнительно данная модель допускает интеграцию не только программного ядра, но и полного интерфейса среды. Четвёртая модель предполагает взаимодействие двух независимых

образовательных платформ. В такой архитектуре Мирера используется как вычислительное и методическое ядро, а внешняя система отвечает за хранение конфиденциальных данных, работу с видеоконтентом и административное сопровождение курсов. Таким образом, разработанная архитектура обеспечивает необходимую гибкость для интеграции с различными цифровыми образовательными средами и может выступать в качестве общей среды взаимодействия, поддерживающей широкий спектр учебных задач. Итоговая архитектура различных моделей интеграции цифровой образовательной платформы Мирера с цифровыми образовательными средами представлена на рисунке, Рисунок 31.

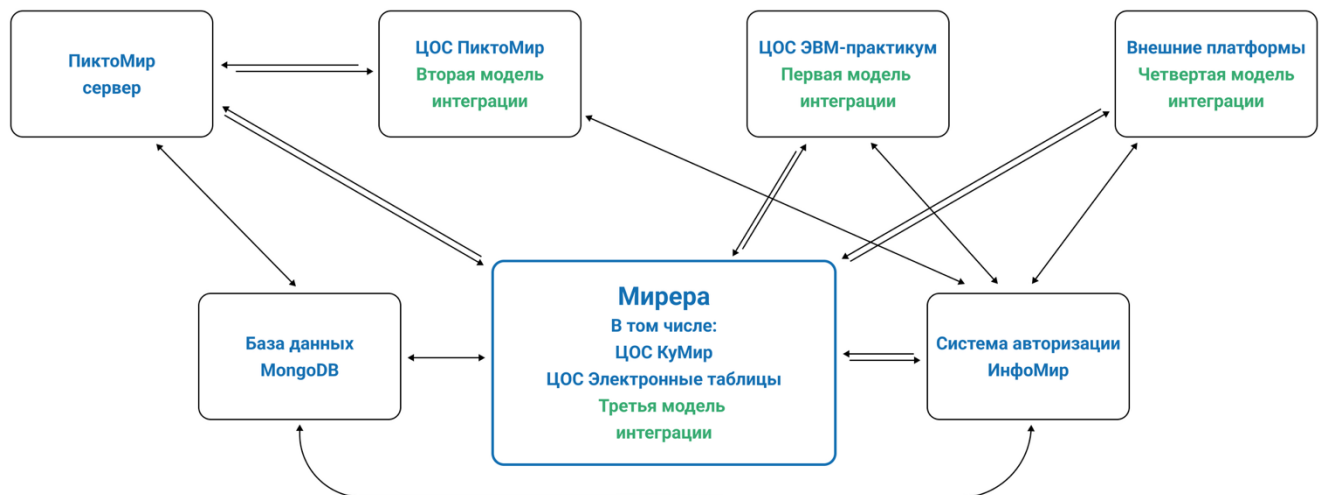


Рисунок 31 – Архитектура различных моделей интеграции ЦОП и ЦОС

4.2.1 Интеграция цифровой образовательной среды ЭВМ-практикум

Примером внешней интеграции с инициацией синхронизации со стороны образовательной платформы является включение в инфраструктуру Миреры цифровой образовательной среды ЭВМ-практикум. Это специализированный симулятор виртуальной вычислительной машины, ориентированный на обучение архитектуре ЭВМ и разработку программ на низкоуровневом языке. Основу функционала составляет моделирование работы регистров, стека, памяти и команд центрального процессора. Обучающийся получает возможность создавать программы, пошагово их выполнять и наблюдать за динамикой изменений в

архитектурных структурах виртуальной машины.

С технической точки зрения взаимодействие реализовано через стандарт авторизации OAuth 2.0, что обеспечивает единый протокол аутентификации и безопасный доступ пользователя Миреры к задачам, которые создаются на платформе и выгружаются в интерфейс ЭВМ-практикума. Среда при этом функционирует автономно: задания передаются в неё из платформы Мирера, после чего студент может отправить код своего решения обратно в платформу с помощью отдельной кнопки. Такой подход позволяет синхронизировать образовательный процесс на стороне платформы, сохраняя при этом в самой среде собственную логику работы и визуальное сопровождение выполнения программы. Пример отображения задания ЭВМ-практикума с загруженным из Миреры условием и возможностью отправки решения представлен на рисунке, Рисунок 32.

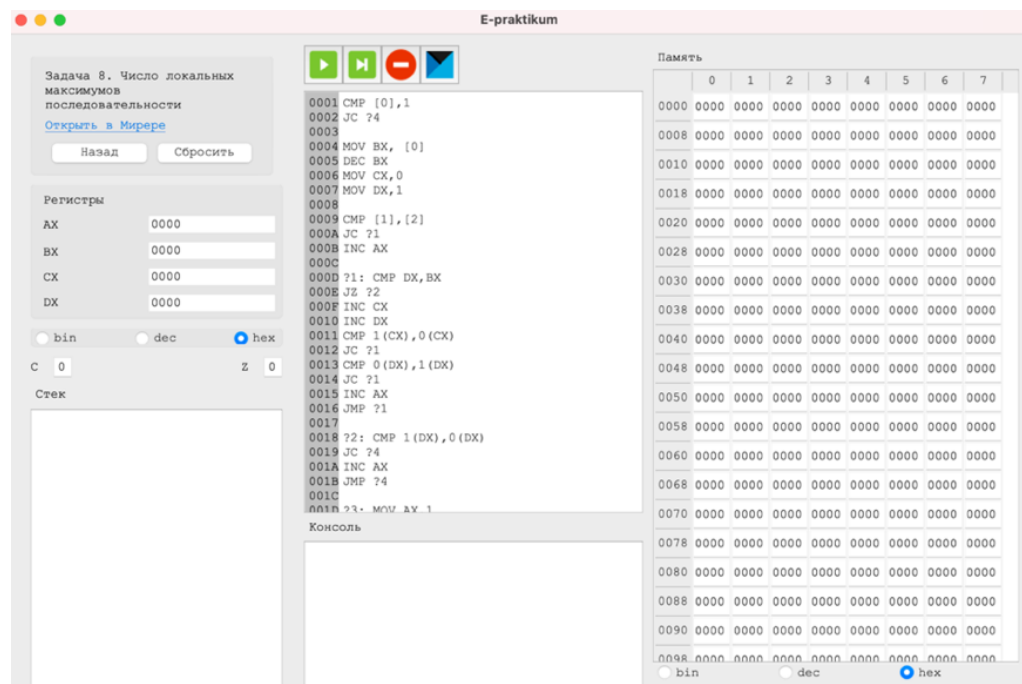


Рисунок 32 – Интеграция ЦОС ЭВМ-практикум

Особенностью интеграции является наличие дополнительного механизма верификации: все полученные решения повторно проверяются встроенным модулем платформы Мирера. Помимо этого, студент может выполнять задания непосредственно в интерфейсе Миреры, где предусмотрено окно для ввода кода на

языке ЭВМ-практикума. Таким образом, данная цифровая среда использует не только внешнюю интеграцию, но и внутреннюю интеграцию ядра ЭВМ-практикума в инфраструктуру платформы. Это позволяет не только посмотреть пошаговый процесс выполнения в отдельной цифровой среде, но и выполнять простые задачи непосредственно внутри платформы без установки стороннего программного обеспечения.

4.2.2 Интеграция цифровой образовательной среды КуМир

Как и в случае возможности разработки кода ЭВМ-практикума напрямую в Мирере, интеграция цифровой образовательной среды КуМир в платформу Мирера реализована по принципу внутренней интеграции с полным встраиванием ядра в образовательную платформу. КуМир – это среда программирования, построенная на базе школьного алгоритмического языка, имеющего наглядный синтаксис и адаптированного для обучения основам алгоритмизации в школьной и вузовской образовательной среде. Исторически КуМир развивался как самостоятельное настольное приложение, ориентированное на офлайн-занятия. Однако переход к дистанционному и смешанному обучению потребовал включения среды в онлайн-инфраструктуру, что и было реализовано в рамках платформы Мирера.

На платформе КуМир интегрирован с использованием встроенного редактора Ace Editor, для которого были разработаны специализированные модули подсветки синтаксиса, стили и поведенческие шаблоны, соответствующие структуре школьного алгоритмического языка. Встроенный компилятор, адаптированный к логике среды КуМир, обеспечивает выполнение программ непосредственно на платформе, без необходимости устанавливать стороннее программное обеспечение. Благодаря этому работают все разработанные механизмы ЦОП Мирера, такие как совместная разработка кода решений задач, антиплагиат, статистика выполнения задач, автоматическая генерация тестов и последующая проверка решений, и многое другое. Помимо этого, преподаватель может создавать шаблоны заданий с защищёнными от редактирования участками,

что существенно упрощает обучение начинающих. Таким образом, КуМир в Мирере представляет собой пример глубокой интеграции, при которой среда утрачивает самостоятельный интерфейс, но сохраняет полную функциональность в составе платформы. Пример отображения кода ЦОС КуМир в ЦОП Мирера представлен на рисунке, Рисунок 33.

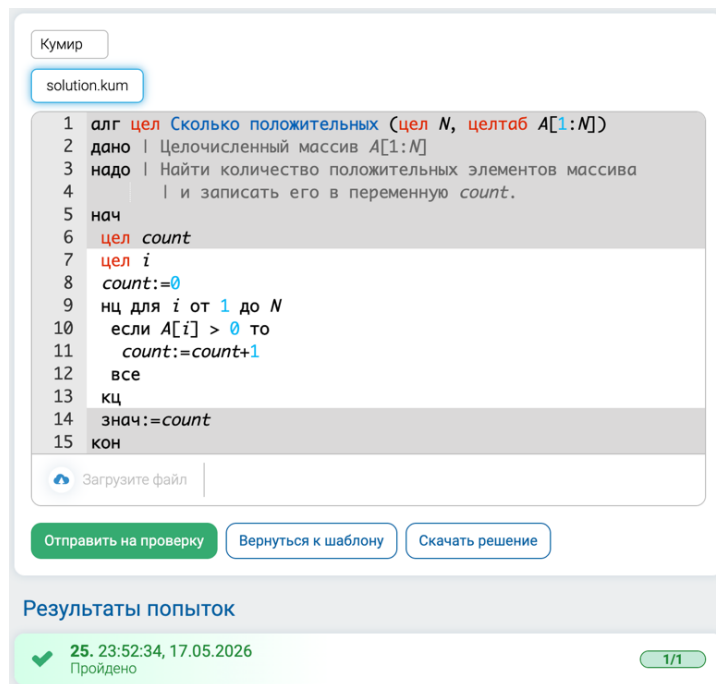


Рисунок 33 – Интеграция ЦОС КуМир

4.2.3 Интеграция цифровой образовательной среды ПиктоМир

Примером внешней интеграции с инициацией синхронизации со стороны образовательной среды является интеграция ЦОС ПиктоМир, представляющей собой бестекстовую цифровую образовательную среду, разработанную для систематического погружения в основы современного программирования. В ПиктоМире обучающиеся самостоятельно составляют программы для управления роботами в виртуальных и реальных обстановках, используя пиктограммы – изображения, представляющие команды роботов и конструкции программирования. ПиктоМир реализует все конструкции современного программирования, что позволяет решать содержательные и интересные задачи, включая командную работу.

Интеграция ПиктоМира в платформу Мирера осуществляется по модели внешнего подключения с использованием протоколов OAuth 2.0, однако, в отличие от ЭВМ-практикума, инициатором синхронизации является сама среда. Все задания, включая структуру курсов, формируются в интерфейсе ПиктоМира. После завершения разработки мира – набора игр и заданий, преподаватель экспортирует их в Миреру, где настраиваются только расписание, видимость и реализуется последующий сбор статистики. Вносить изменения в уже выгруженные задания из интерфейса платформы невозможно – любые корректировки должны быть сделаны в самой среде ПиктоМир, после чего производится повторная выгрузка и обновление контента в Мирере.

Интерфейс ПиктоМира интегрирован в Миреру посредством встроенного iframe окна на странице задачи, что позволяет обучающемуся выполнять задания в привычной визуальной среде, не покидая пределов цифровой образовательной платформы, Рисунок 34. По завершении работы результаты автоматически передаются обратно в Миреру, где хранятся в профиле пользователя и могут быть использованы для анализа преподавателем.

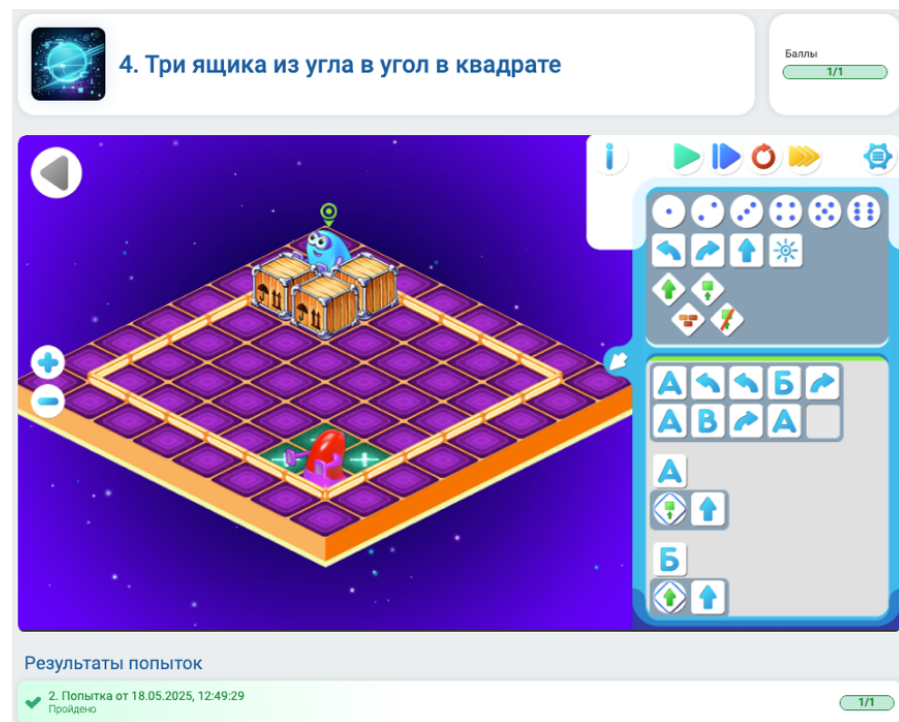


Рисунок 34 – Интеграция ЦОС ПиктоМир

Дополнительной гарантией достоверности выполнения является встроенное в платформу Мирера дублирующее проверяющее ядро ПиктоМира. Оно функционирует без графического интерфейса и предназначено для верификации решений, отправленных студентами. Это повышает надёжность оценки, позволяет выявлять некорректные или модифицированные данные и обеспечивает независимый контроль качества прохождения заданий. Таким образом, интеграция ПиктоМира представляет собой вариант асинхронного взаимодействия, при котором инициатива синхронизации исходит от внешней среды, а сама схема работы остаётся согласованной с архитектурой платформы Мирера и включает встроенные механизмы контроля.

4.2.4 Интеграция цифровой образовательной среды Электронные таблицы

Примером внутренней интеграции с полным встраиванием интерфейса в Миреру является ЦОС Электронные таблицы со встроенной автоматической проверкой решений. До недавнего времени основным стандартом в области электронных таблиц являлось программное обеспечение Microsoft Excel. В целях обеспечения системного формирования компетенций по работе с электронными таблицами, в том числе в условиях импортозамещения, в рамках цифровой образовательной платформы Мирера был разработана и внедрена цифровая образовательная среда, обеспечивающая полноценную работу с электронными таблицами в составе учебных заданий. Реализация данной ЦОС позволяет создавать интерактивные задания, которые соответствуют требованиям школьных программ и экзаменационных форматов, с автоматизированной проверкой и встроенной интеграцией с другими основными механизмами взаимодействия участников образовательного процесса, реализованных в цифровой образовательной платформе Мирера. ЦОС Электронные таблицы работает без обращения к сторонним API и библиотекам, а также без необходимости установки дополнительного программного обеспечения на устройства пользователей – она полностью функционирует в браузере пользователя, вне зависимости от операционной системы и локального ПО.

Ключевым компонентом реализации данной ЦОС стала поддержка набора базовых табличных формул, включающий, в частности, SUM, MAX, MIN, IF, AND, OR, COUNTIF, SUMIF и AVERAGEIF. Для обеспечения корректной работы всех выражений реализован рекурсивный механизм интерпретации вложенных формул, состоящий в обработке строковых выражений и поэтапном вычислении вложенных подформул, который позволяет вычислять значения даже при наличии сложных, иерархически вложенных логических конструкций. На рисунке, Рисунок 35, приведен заблокированный шаблон с исходными данными задачи, а также ячейки в обычном состоянии и ячейку в состоянии редактирования с примером формулы.

	A	B	C	D	E	F
1	округ	фамилия	предмет	балл		
2	С	Ученик 1	физика	240	=IF(AND(D2>600;C2="информатика");D2;0)	0
3	В	Ученик 2	физкультура	782	0	0
4	Ю	Ученик 3	биология	361	0	0
5	СВ	Ученик 4	обществознание	377	0	
6	ЮЗ	Ученик 5	информатика	542	0	0
7	В	Ученик 6	физкультура	606	0	0
8	СЗ	Ученик 7	информатика	804	804	1
9	ЮЗ	Ученик 8	биология	118	0	0
10	Ю	Ученик 9	обществознание	938	0	0
11	СВ	Ученик 10	обществознание	115	0	0
12	ЮЗ	Ученик 11	физкультура	426	0	0
13	ЮВ	Ученик 12	физкультура	448	0	0
14	СЗ	Ученик 13	физкультура	209	0	0
15	ЮЗ	Ученик 14	информатика	771	771	1

+ ... Лист1

Загрузите файл | Выберите или переместите файл сюда

Отправить на проверку | Вернуться к шаблону | Скачать решение

Результаты попыток

❌ 5. 12:16:37, 23.01.2026
Значение в ячейке F5 не соответствует ответу

0/1

Рисунок 35 – Пример задачи на электронные таблицы

4.2.5 Распределенная архитектура интеграции с внешними образовательными платформами

Интеграция ЦОП Мирера с независимыми образовательными платформами, функционирующими в рамках собственной инфраструктуры и логики, представляет собой отдельный класс задач. Такой формат взаимодействия актуален

для ситуаций, когда одна платформа сосредоточена на методической и вычислительной части обучения, а другая – на управлении и хранении мультимедийного контента и конфиденциальной информации. В рамках подобной архитектуры ЦОП Мирера выполняет функции учебного ядра: именно здесь размещаются и исполняются задания, реализуется автоматическая проверка, ведётся учёт результатов, тогда как внешняя платформа осуществляет управление курсами, пользователями и образовательным контентом.

Взаимодействие организуется на основе стандартизированных протоколов, включая OAuth 2.0 для авторизации и защищённого обмена данными. Все запросы инициируются со стороны внешней платформы: в случае регистрации пользователя, привязки курса, подключения групп или запроса статистики, она обращается к открытому API платформы Мирера. При первом переходе пользователя на внешнюю платформу инициируется всплывающее окно авторизации в Миреру, в ходе которой формируется токен доступа, действующий в течение одного месяца, а также краткосрочные токены на каждую сессию. Такая система позволяет обеспечить не только единый вход, но и высокую степень безопасности при межплатформенном взаимодействии.

Функциональная связка между курсами и группами реализуется через механизм синхронизации: преподаватель создаёт образовательную структуру в Мирере, после чего во внешней платформе выбирает соответствующие курсы и группы для сопоставления. После настройки связки преподаватель получает в интерфейсе внешней платформы доступ к информации о контекстах из Миреры, включая тип занятия, время проведения и связанные учебные группы, а данные автоматически синхронизируются с календарём и журналом преподавателя.

Для студентов интеграция организована аналогичным образом: после регистрации их данные передаются в Миреру, где автоматически создаётся профиль и выполняется подключение к учебным группам. В интерфейсе курса студент получает доступ к контекстам, информации о сроках выполнения и статусе прохождения, а результаты выполнения заданий автоматически возвращаются во внешнюю платформу и отображаются в рамках общего учебного трека.

Особенностью архитектуры является разделение ответственности между системами: Мирера отвечает за методическое наполнение, проверку знаний и накопление цифрового следа, а внешняя платформа – за административное сопровождение, работу с конфиденциальными данными, видеоконтент и отчетность.

4.3 Экспериментальная апробация и оценка эффективности разработанных модулей в условиях цифровой образовательной платформы

Проведенная интеграция разработанных модулей в архитектуру цифровой образовательной платформы Мирера позволила перейти от теоретического обоснования и алгоритмического описания к практической проверке их работоспособности в условиях реального образовательного процесса. В отличие от изолированного тестирования отдельных компонентов, апробация осуществлялась в рамках функционирующей системы, включающей реальные учебные курсы, преподавателей и обучающихся, что обеспечило высокую степень достоверности получаемых результатов.

Ключевой особенностью проведенной апробации является комплексный характер внедрения: модули функционировали не как независимые инструменты, а как взаимосвязанные компоненты, обменивающиеся данными через общую инфраструктуру платформы. Это позволило оценить не только их индивидуальную эффективность, но и влияние на сквозные процессы образовательной деятельности, включая сбор и обработку цифрового следа, автоматизацию проверки заданий, поддержку обучающихся и аналитическое сопровождение учебного процесса.

Экспериментальная эксплуатация проводилась на выборке учебных дисциплин различного профиля, включая программирование, математические и прикладные курсы, а также дисциплины, требующие развернутых текстовых ответов. В рамках апробации фиксировались как количественные метрики (временные затраты, частота использования, точность алгоритмов, нагрузка на преподавателя), так и качественные характеристики (изменение поведения обучающихся, степень вовлечённости, согласованность оценивания,

интерпретируемость результатов). Отдельное внимание уделялось сравнению реализованных решений с традиционными подходами, применяемыми до внедрения модулей. Это позволило выявить не только абсолютные показатели эффективности, но и относительные изменения, отражающие вклад автоматизации в оптимизацию образовательного процесса. При этом анализ проводился с учётом различий в форматах обучения (очный, дистанционный, гибридный), что обеспечило проверку универсальности предложенных решений.

Важно подчеркнуть, что полученные в ходе апробации результаты характеризуют не частный случай внедрения в конкретную платформу, а отражают более общие закономерности функционирования интегрированных интеллектуальных модулей в цифровых образовательных системах. Это связано с тем, что архитектурные и алгоритмические принципы, заложенные в основу разработанных решений, изначально ориентированы на платформенную независимость и масштабируемость.

4.3.1 Оценка эффективности модуля автоматизированного учёта посещаемости

Внедрение разработанного модуля учёта посещаемости позволило получить количественно измеримые эффекты, отражающие как повышение эффективности организационных процессов, так и улучшение качества собираемых данных. В частности, по результатам апробации системы в учебных группах различной численности было зафиксировано существенное сокращение временных затрат преподавателя на фиксацию посещаемости. Если при использовании традиционных методов этот процесс занимал в среднем до 5 минут на одну сессию для группы из 20-30 человек, то применение автоматизированных механизмов (геолокация и QR-коды) сократило это время до 30-60 секунд, включая единичные случаи ручной корректировки. Таким образом, относительное сокращение затрат времени составляет примерно 80-90%, что особенно важно в условиях больших аудиторий.

Кроме того, было отмечено сокращение количества ошибок и неточных записей. В традиционных сценариях имели место случаи фиктивных записей о посещаемости отсутствующих студентов, вызванные человеческой ошибкой или передачей информации через третьих лиц. После внедрения системы частота таких несоответствий снизилась до единичных случаев, причем большинство из них выявлялось автоматически благодаря встроенным механизмам проверки и контрольному показателю подсчета количества людей на фотографии. Это значительно повысило точность журналов посещаемости и снизило нагрузку на преподавателей, связанную с последующей проверкой и исправлением данных.

В формате дистанционного обучения одним из ключевых результатов стало повышение репрезентативности показателей посещаемости. В отличие от ранее использовавшегося подхода, основанного исключительно на факте входа в систему на занятии, внедренная система учитывает фактическое время присутствия студента и степень его вовлеченности. Анализ показал, что доля студентов, формально зарегистрированных на занятии, но не демонстрирующих фактической активности, может достигать 30-40% при использовании упрощенных методов отслеживания. В рамках предложенной архитектуры такие случаи либо не учитываются как полное посещение, либо получают пропорционально уменьшенное количество баллов, что приводит к более точному отражению вовлеченности студентов.

Наконец, интеграция всех видов активности в единую систему обеспечила сквозную сопоставимость данных между очным, дистанционным и гибридным форматами. Это, в свою очередь, заложило основу для последующего использования накопленных данных в задачах образовательной аналитики, включая построение индивидуальных траекторий обучения. Таким образом, полученные количественные результаты подтверждают, что разработанный модуль не только преодолевает ограничения традиционных подходов, но и устанавливает качественно новый уровень надежности и операционной эффективности процессов учета посещаемости. Итоговые сводные результаты апробации модуля представлены в таблице, Таблица 6.

Таблица 6 – Результаты апробации модуля автоматизированного учета посещаемости

Ключевой показатель	Значение до внедрения / Традиционный подход	Значение после внедрения	Достигнутый эффект
Время на фиксацию посещаемости (группа 20-30 человек)	До 5 мин. (ручная переключка)	30-60 сек. (автоматическая геолокация и QR-коды)	Сокращение временных затрат на 80-90%
Доля недостоверных записей	До 15-20% случаев	Единичные случаи, выявляются автоматически	Повышение достоверности журнала посещаемости
Доля студентов с формальной регистрацией, но без фактической активности	30-40% (при учёте только по факту входа)	≈0% (гражданская система баллов за реальное время присутствия)	Повышение репрезентативности данных о вовлечённости

4.3.2 Оценка эффективности модуля обнаружения заимствований в программных решениях

Результаты апробации разработанного алгоритма обнаружения заимствований подтвердили его практическую применимость в условиях цифровой образовательной платформы и продемонстрировали качественное улучшение характеристик процесса проверки программных решений по сравнению с традиционными подходами. В рамках пилотного внедрения, проведённого на выборке учебных заданий по программированию, было зафиксировано существенное снижение трудозатрат преподавателя на анализ решений обучающихся.

Особого внимания заслуживает влияние системы на надежность итоговой оценки результатов обучения. При традиционном подходе, основанном на частично ручной проверке, сохраняется высокая вероятность как пропуска плагиата, так и субъективной интерпретации совпадений. Внедрение алгоритма позволило стандартизировать процедуру проверки и обеспечить единообразие критериев оценки, что особенно важно при обучении больших групп. Косвенно это подтверждается снижением числа апелляций со стороны обучающихся, связанных с несогласием с результатами проверки. Еще одним результатом является улучшение интерпретируемости итоговых данных. В отличие от агрегированных методов, которые предоставляют лишь общий показатель сходства, разработанный

алгоритм генерирует подробную аннотацию совпадающих фрагментов с указанием вероятностного показателя для каждой строки.

Для экспериментальной оценки проведено сравнение с базовым методом, основанным на формировании отпечатков токенизированной программы. На наборе из 1173 пар программ с известным фактом заимствования разработанный метод обеспечил увеличение F1-меры с 92.73% до 99.19% и снижение доли пропущенных заимствований с 13.55% до 0.85%. Результаты оценки устойчивости методов к различным способам маскировки заимствований представлены в таблице, Таблица 7. Наиболее существенное преимущество разработанного метода получено при одновременном применении нескольких способов маскировки: доля выявленных заимствований увеличилась с 58.47% до 96.19%. В остальных рассмотренных случаях также наблюдается повышение доли выявленных заимствований, что подтверждает устойчивость разработанного метода к структурным и синтаксическим изменениям исходного кода.

Таблица 7 – Результаты апробации модуля обнаружения заимствований в программных решениях

Ключевой показатель	Метод отпечатков на токенизированной программе (baseline)	Разработанный модуль обнаружения заимствований
F1-мера	92.73%	99.19%
Доля пропущенных структурных заимствований	13.55%	0.85%
Доля обнаружения при переименовании	92.45%	100.00%
Доля обнаружения при форматировании	99.57%	100.00%
Доля обнаружения при вставке лишнего кода	99.58%	100.00%
Доля обнаружения при замене синтаксических конструкций	74.85%	99.39%
Доля обнаружения при перестановке кода	97.85%	100.00%
Доля обнаружения при применении комплексной маскировки	58.47%	96.19%

4.3.3 Оценка эффективности модуля автоматизированной семантической проверки текстовых ответов

В ходе апробации разработанного модуля автоматизированной проверки текстовых ответов оценивались качество классификации, трудоёмкость проверки и согласованность результатов оценивания. Для оценки качества классификации проведено экспериментальное исследование на 737 ответах студентов по дисциплинам технического профиля, предусматривающим развернутые текстовые ответы. Система автоматически классифицировала 87.25% ответов, при этом среди автоматически проверенных ответов ошибок первого и второго рода не допущено, что соответствует F1-мере 100%. Остальные 12.75% ответов были отнесены к неопределённым и направлены на дополнительную проверку. В сравнении с системой [16], которая единственная работает с русскоязычными текстами и для которой F1-мера составила 89.68%, разработанный подход обеспечивает более надёжную автоматическую классификацию за счёт передачи неопределённых случаев на дополнительную проверку.

Использование модуля позволяет существенно сократить временные затраты на проверку работ. При традиционном подходе, основанном на ручной проверке и последующем переносе результатов в электронные системы, среднее время обработки одного ответа составляло порядка 2-3 минут. При использовании модуля первичная оценка формируется автоматически, а участие преподавателя требуется только для пограничных случаев, доля которых, по результатам наблюдений, не превышает 10-15% от общего числа ответов. Таким образом, общее время проверки сокращается в несколько раз, что особенно важно при работе с большими группами обучающихся.

С точки зрения согласованности оценивания использование формализованного алгоритма сравнения синтаксических деревьев снижает влияние субъективных факторов и обеспечивает воспроизводимость результатов при одинаковых входных данных. Предлагаемый подход обеспечивает единообразие критериев оценки, что особенно важно при проверке большого

количества похожих заданий. Механизм пороговой классификации позволяет оптимизировать распределение рабочей нагрузки между автоматизированной системой и преподавателем: однозначные ответы классифицируются автоматически, а неоднозначные направляются на дополнительный анализ. Такой подход обеспечивает баланс между автоматизацией и контролем качества, устраняя необходимость в полной ручной проверке всех заданий. В совокупности результаты подтверждают сокращение временных затрат, повышение надёжности и воспроизводимости процедуры оценки текстовых ответов. Итоговые результаты апробации модуля представлены в таблице, Таблица 8.

Таблица 8 – Результаты апробации модуля автоматизированной семантической проверки текстовых ответов

Ключевой показатель	Значение до внедрения / Традиционный подход	Значение после внедрения	Достигнутый эффект
F1-мера автоматической классификации	89.68% (в единственной системе, автоматически проверяющей ответы на русском языке)	100% (12.75% переданы на ручную проверку преподавателем)	Повышение надёжности проверки за счёт передачи неопределённых случаев преподавателю
Среднее время проверки одного развёрнутого ответа	2-3 мин. (ручная проверка, без учёта переноса результата)	≈0 мин. для 85-90% ответов (автоматическая классификация); 2-3 мин. для 10-15% пограничных случаев	Сокращение суммарного времени проверки в несколько раз
Воспроизводимость и согласованность оценки	Субъективная, зависит от преподавателя и момента проверки	Абсолютная, обеспечивается формальным алгоритмом сравнения деревьев	Повышение объективности и прозрачности оценивания
Доля ответов, требующих ручной проверки преподавателем	100% (классическая ручная проверка)	10-15% (попадающих в зону неопределённости между порогами)	Оптимизация распределения нагрузки между автоматической системой и преподавателем

4.3.4 Оценка эффективности применения нейросетевых ассистентов в образовательном процессе

Экспериментальное тестирование разработанного решения проводилось в рамках курса по программированию с использованием цифровой образовательной платформы. Особое внимание уделялось сравнению подгрупп студентов в пределах одного курса, что позволило исключить влияние различий в их исходном

уровне подготовки. В ходе тестирования регистрировались все взаимодействия студентов с нейросетевым помощником, параметры генерируемых подсказок и последующие результаты выполнения заданий. Это позволило получить подробные данные об использовании системы и ее влиянии на процесс обучения.

Анализ экспериментальных данных показал, что частота обращений к помощнику коррелирует со сложностью задач, измеряемой средним временем их решения. Студенты редко используют систему при выполнении тривиальных упражнений и обращаются к ней преимущественно в ситуациях, когда сталкиваются с реальными трудностями. Это свидетельствует о сознательном использовании инструмента и опровергает предположение о том, что он используется исключительно в качестве средства упрощения учебной деятельности.

Особый интерес представляет анализ поведения студентов в рамках одной учебной группы. В ходе пилотного исследования были выявлены различные стратегии использования помощника, отличающиеся по частоте и характеру запросов. Результаты показали, что студенты, активно использующие нейросетевого помощника, достигают более высоких итоговых оценок по сравнению с теми, кто использует его лишь изредка или не использует вовсе. Медианный рост итоговых образовательных показателей у активно использующих нейроассистента студентов составил 15.92%. В то же время наиболее выраженный положительный эффект наблюдается у студентов, которые регулярно используют подсказки для устранения блокирующих ошибок и продолжают работать над задачей до тех пор, пока не получают правильный результат, Рисунок 36. При интерпретации данных важно отметить, что повышение успеваемости сопровождается увеличением среднего числа попыток сдачи решений и времени, затрачиваемого на выполнение заданий. Обучающиеся реже бросают решение задачи после первой неудачи и чаще продолжают работу, используя подсказки системы для устранения ошибок шаг за шагом. Таким образом, увеличение количества попыток рассматривается как показатель более высокой вовлеченности и устойчивости.

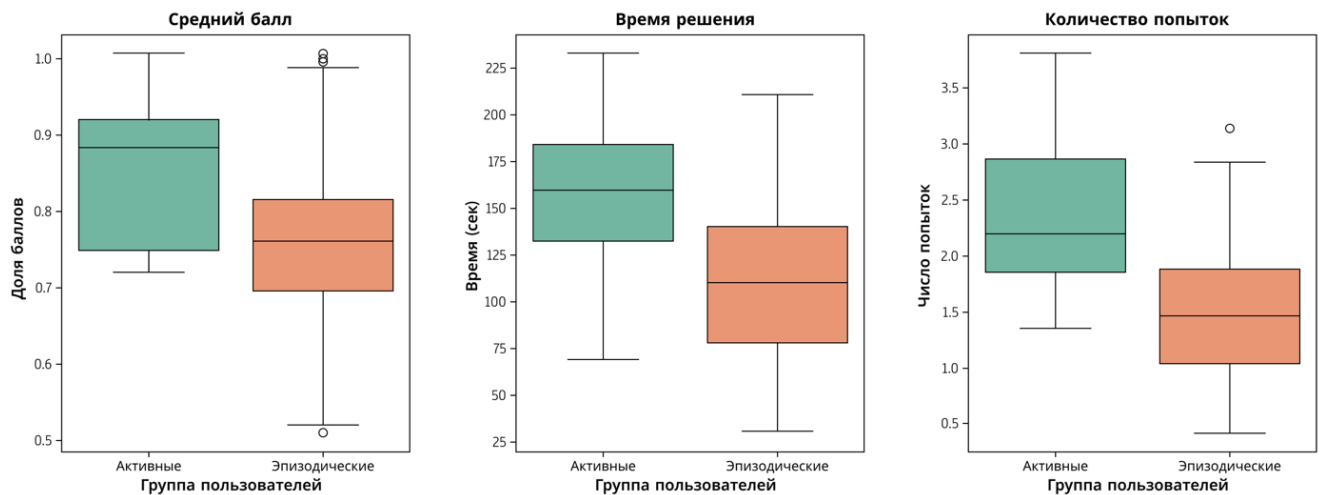


Рисунок 36 – Оценка влияния использования нейросетевого ассистента на распределение образовательных метрик среди обучающихся

С точки зрения функционирования платформы, одним из ключевых преимуществ является снижение пиковых нагрузок на преподавателей, что позволяет им уделять больше времени индивидуальной работе со студентами, испытывающими системные трудности. Количественная оценка показывает, что значительная доля вопросов, которые в традиционной модели обучения адресовались бы преподавателю, успешно обрабатывается нейросетевым ассистентом. Среднее количество обращений к нейроассистенту за одно занятие составило $\sim 3,74$ на одного взрослого студента, который умеет интерпретировать сообщения тестирующей системы, и $\sim 5,87$ на школьника, который еще недостаточно владеет английским языком и навыком чтения документации, что при группе из 28 человек соответствует пиковой нагрузке на преподавателя от 104 до 165 индивидуальных консультаций.

В совокупности результаты экспериментальной апробации подтверждают, что разработанный нейросетевой ассистент не только снижает нагрузку на преподавателей, но и способствует изменению учебного поведения обучающихся. Система поддерживает процесс активного поиска решения, снижает влияние психологических барьеров при обращении за помощью и повышает устойчивость обучающихся при работе с трудными заданиями.

Существенная часть временных затрат преподавателя также связана с

регулярной разработкой заданий на основе лекционного материала, что необходимо выполнять после каждой темы для поддержания актуальности и согласованности учебного контента. В традиционном подходе данный процесс включает последовательный анализ лекции, выделение ключевых элементов, подлежащих проверке, формализацию условий задач, а также их последующее структурирование и перенос в цифровую образовательную платформу. В зависимости от объема и сложности материала суммарное время выполнения этих операций составляет в среднем от 40 до 90 минут на одну лекцию, при этом значительная часть времени приходится на рутинные операции форматирования и адаптации заданий под требования системы. Использование нейросетевого ассистента позволяет автоматизировать данный процесс за счет генерации заданий, согласованных с содержанием лекции. В результате суммарное время подготовки заданий сокращается до 5-10 минут с учетом этапа верификации, что соответствует снижению трудозатрат в среднем в 5-8 раз. При этом уменьшается количество технических ошибок, связанных с переносом и форматированием контента. Итоговые результаты апробации нейросетевых ассистентов представлены в таблице, Таблица 9.

Таблица 9 – Результаты апробации нейросетевых ассистентов

Ключевой показатель	Значение до внедрения / Традиционный подход	Значение после внедрения	Достигнутый эффект
Пиковая нагрузка на преподавателя по консультациям (группа 28 человек)	104-165 индивидуальных запросов (при массовом обучении программированию)	Нагрузка с преподавателя снимается, запросы обрабатываются ассистентом	Полное снятие пиковой рутинной нагрузки по типовым ошибкам с преподавателя
Успеваемость студентов, активно использующих ассистента, по сравнению с не использующими	не нормировалась	Медианный рост итоговых образовательных показателей на 15.92%.	Повышение вовлечённости и настойчивости при решении сложных задач
Время на подготовку учебных заданий по материалам одной лекции	40-90 мин. (анализ лекции, формализация условий, перенос в платформу)	5-10 мин. (автоматическая генерация заданий с последующей верификацией)	Снижение временных затрат в 5-8 раз

4.3.5 Оценка эффективности модуля аналитики цифрового следа обучающихся

Результаты внедрения разработанного модуля показали, что ключевое улучшение связано не столько с автоматизацией сбора данных как таковой, сколько с переходом к их системной аналитической обработке и интерпретации в наглядной форме. В исходной конфигурации образовательной платформы данные о посещаемости, оценках и результатах выполнения заданий уже регистрировались в цифровом виде, однако их использование ограничивалось в основном ситуативным просмотром и ручным анализом, что значительно снижало их практическую ценность для принятия решений.

Внедрение аналитического модуля позволило объединить разрозненные показатели в единую картину и визуализировать их с различной степенью детализации. В частности, это дало возможность перейти от анализа отдельных значений к анализу распределений и тенденций в показателях. Например, использование boxplot-графиков позволило выявить не только средние значения, но и изменчивость результатов внутри группы, включая наличие выбросов и асимметрию распределения. В некоторых случаях это привело к пересмотру интерпретации результатов: при сопоставимых средних баллах различия в межквартильном размахе достигали 20-25%, что указывало на неоднородность усвоения материала внутри группы. Кроме того, наблюдалось повышение эффективности сравнительного анализа. Если ранее сравнение результатов двух групп учащихся или отдельных учащихся требовало ручной обработки данных и занимало значительное время, то в рамках внедренного решения эти операции выполняются посредством визуализации на единой шкале.

Существенным результатом стало также появление возможности анализа временной динамики показателей. Визуализация собираемых метрик по мере прохождения курса позволила выявлять моменты снижения активности студентов и соотносить их с конкретными темами или контестами. Это обеспечило более точную локализацию проблемных участков курса.

Таким образом, внедрение модуля позволило перейти от фрагментарного использования отдельных показателей к комплексному анализу образовательных данных, включающему агрегирование, визуализацию и интерпретацию метрик. Это, в свою очередь, обеспечило повышение обоснованности педагогических выводов и позволило использовать данные цифрового следа не только как средство фиксации результатов, но и как инструмент системного анализа образовательного процесса. Итоговые результаты апробации модуля представлены в таблице, Таблица 10.

Таблица 10 – Результаты апробации модуля аналитики цифрового следа

Ключевой показатель	Значение до внедрения / Традиционный подход	Значение после внедрения	Достигнутый эффект
Характер аналитической обработки данных	Фрагментарный, ручной анализ отдельных показателей	Системный, автоматизированный анализ распределений, динамики и взаимосвязей метрик	Переход от фиксации результатов к инструменту системного анализа образовательного процесса
Способ выявления различий между группами по успеваемости	Ручная обработка и сравнение данных	Автоматическая визуализация в едином масштабе (box-plot, линейные графики)	Повышение обоснованности педагогических выводов и возможность тестирования гипотез
Возможность прогнозирования академических результатов студента	Не реализована	Реализована на основе сигнатур временных рядов цифрового следа	Создание основы для адаптивного управления обучением и раннего выявления студентов группы риска

4.3.6 Оценка эффективности сигнатурных методов при анализе и прогнозировании образовательных траекторий

Анализ образовательной траектории студента предполагает учет многомерных временных характеристик, отражающих его учебную активность. К числу ключевых параметров, формирующих такую траекторию, относятся показатели академической успеваемости, число предпринятых попыток и потраченного времени решения на выполнения заданий, степень схожести решений с эталонными (в контексте оценки системы антиплагиата), а также данные о посещаемости. Эти параметры организованы в виде временного ряда, при этом

каждый отдельный контекст включает в себя вложенную временную структуру, описывающую последовательность попыток решения. Последние, в свою очередь, содержат данные о времени выполнения, статусе решения, набранных баллах и других значимых характеристиках, что позволяет более точно описывать индивидуальные особенности поведения обучающегося в рамках конкретного этапа образовательного процесса. Для представления таких высокоразмерных временных данных в формат, пригодный для обработки алгоритмами машинного обучения, используются сигнатуры временных рядов. Этот подход позволяет эффективно кодировать как значения признаков, так и их взаимосвязи во времени, сохраняя информацию о внутренних зависимостях между параметрами образовательной активности.

Сложность экспериментальной верификации обучающих стратегий обусловлена ограниченностью объема данных: накопление репрезентативной выборки требует многолетних наблюдений, при этом ежегодный поток студентов зачастую недостаточен. Чтобы компенсировать это ограничение, можно использовать генеративные модели, основанные на представлении сигнатур, которые позволяют синтезировать дополнительные траектории, сохраняющие статистические характеристики исходных данных. Такие синтетические выборки можно использовать для обучения нейронных сетей и других моделей.

Основная цель данной модели – предсказать дальнейший ход образовательной траектории студента. Это позволяет сформулировать важные педагогические гипотезы, такие как вероятность снижения академической успеваемости или необходимость срочной поддержки со стороны преподавателя. Для построения модели определяется окно наблюдения, охватывающее последние оценки студента. На основе этих данных формируется временной ряд, из которого вычисляется сигнатура. Целевым значением для модели является следующий элемент ряда, что позволяет обучить регрессионную модель для прогнозирования последующего состояния траектории.

Важно отметить, что обучающая и целевая выборки в данной задаче являются многомерными и включают показатели успеваемости, посещаемости и

другие характеристики. Преимущество сигнатур заключается в их способности выявлять сложные зависимости между признаками, например, корреляцию между снижением посещаемости и увеличением уровня заимствований, либо между числом попыток и качеством решений.

Работа модели продемонстрирована на графике, где первые контесты используются как база для построения прогноза, после чего окно последовательно сдвигается, формируя цепочку предсказаний, Рисунок 37. При этом модель использует фактические данные в режиме реального времени, что делает её применимой в процессе текущего сопровождения учебной деятельности. Альтернативно, при недоступности новых данных, можно использовать сгенерированные предсказания в качестве входа для следующего шага, однако следует учитывать эффект накопления ошибки при каждом смещении окна. Оценка качества работы модели включает не только традиционные регрессионные метрики, но и способность к корректной классификации изменения образовательного статуса студента. Однако в реальной практике педагог может адаптировать свои действия на основе предсказаний модели, чем изменяет итоговое поведение студентов, в связи с чем невозможно использование некоторых метрик при последующей валидации.



Рисунок 37 – Пример работы модели предсказания траектории студента с помощью сигнатурных методов

Для экспериментальной оценки эффективности сигнатурного представления проведено сравнение разработанной сигнатурной модели с базовой регрессионной моделью, непосредственно использующей значения временного ряда без сигнатурного преобразования. Эксперименты выполнены при размерах окна наблюдения в 5, 10, 15 и 20 элементов, в качестве основной метрики использовалась средняя абсолютная ошибка (MAE) на тестовой выборке, рассчитанная по компоненте, соответствующей показателю набранных баллов, как ключевому показателю образовательной траектории. Результаты представлены в таблице, Таблица 11.

Таблица 11 – Результаты сравнения базовой и сигнатурной моделей прогнозирования образовательной траектории

Размер окна	Baseline model, MAE	Signature model, MAE	Снижение MAE
5	0.2250	0.2052	8.80%
10	0.2164	0.2006	7.30%
15	0.2091	0.1933	7.56%
20	0.2027	0.1861	8.19%

Для всех исследованных размеров окна сигнатурная модель демонстрирует меньшую ошибку прогнозирования по сравнению с базовой. Снижение MAE составило от 7.30% до 8.80%. При увеличении размера окна наблюдения ошибка обеих моделей в целом снижается, однако использование модели с меньшим окном позволяет получать прогноз на более раннем этапе обучения. Поэтому на практике на начальных этапах может применяться модель с небольшим окном наблюдения, после чего последовательно заменяться моделью с большим окном и более высокой точностью, что позволяет сочетать возможность более раннего прогнозирования с повышением точности по мере накопления данных индивидуальной образовательной траектории.

Выводы по главе 4

В четвертой главе рассмотрены вопросы интеграции разработанных интеллектуальных модулей в единую цифровую образовательную платформу, а также проведена их экспериментальная апробация в условиях реального образовательного процесса. Полученные результаты позволили перейти от разработки отдельных алгоритмических и программных решений к их комплексному внедрению и оценке эффективности в составе функционирующей системы.

В ходе исследования предложена архитектура цифровой образовательной платформы, обеспечивающая интеграцию различных модулей и внешних образовательных сред в единое информационное пространство. Показано, что использование модульного и распределённого подходов позволяет обеспечить масштабируемость системы, гибкость её настройки и возможность поэтапного расширения функциональности без нарушения целостности архитектуры. Реализованные механизмы интеграции обеспечивают взаимодействие как с внутренними компонентами платформы, так и с внешними образовательными сервисами, что позволяет формировать единую цифровую образовательную среду.

Рассмотрены архитектурные подходы к интеграции внешних образовательных сред и модулей различного типа. Показано, что использование как внутренней, так и внешней интеграции, а также решений распределенной архитектуры позволяет включить специализированные учебные среды в единое образовательное пространство без ущерба для их функциональности. Это обеспечивает масштабируемость платформы и её адаптируемость к различным образовательным сценариям и дисциплинам.

Экспериментальная апробация разработанных модулей продемонстрировала их функциональность и практическую значимость в решении задач автоматизации образовательного процесса. Установлено, что модуль учета посещаемости повышает точность и полноту фиксации учебной активности студентов в различных формах обучения. Модуль обнаружения плагиата демонстрирует

устойчивость к различным методам модификации программного кода и повышает объективность контроля самостоятельного выполнения заданий. Модуль семантической проверки текстовых ответов позволяет автоматизировать оценку заданий открытого типа на основе их семантического содержания, тем самым расширяя спектр задач, поддающихся автоматической проверке.

Показано, что использование помощников на основе нейронных сетей способствует повышению доступности образовательной поддержки для студентов и снижает рутинную нагрузку на преподавателей при подготовке и организации учебного процесса. Установлено, что интеграция таких помощников в образовательную платформу позволяет предоставлять контекстную помощь, адаптированную к текущему этапу обучения и содержанию учебного материала.

Результаты апробации модуля анализа цифрового следа демонстрируют возможность эффективного мониторинга образовательной деятельности, выявления поведенческих паттернов студентов и построения прогнозных оценок их академической успеваемости. Применение методов сигнатурного анализа временных рядов подтвердило их актуальность для моделирования образовательных траекторий и повысило точность прогнозов по сравнению с традиционными статистическими подходами.

В целом, результаты апробации подтверждают, что интеграция разработанных интеллектуальных модулей в единую цифровую образовательную платформу обеспечивает комплексную автоматизацию ключевых образовательных процессов, повышает объективность оценки знаний, расширяет аналитические возможности и способствует повышению эффективности образовательного процесса.

ЗАКЛЮЧЕНИЕ

В рамках диссертационного исследования были успешно решены поставленные задачи, направленные на разработку универсальных интеллектуальных модулей, обеспечивающих высокий уровень автоматизации образовательного процесса. Полученные результаты направлены на снижение рутинной нагрузки преподавателей, повышение прозрачности и управляемости учебного процесса, а также создание условий для анализа и сопровождения индивидуальных образовательных траекторий обучающихся. Основные результаты работы заключаются в следующем:

1. Проведён комплексный анализ современных технологий автоматизации образовательного процесса, включая архитектурные принципы построения и разработки цифровых образовательных платформ с отдельными интеллектуальными модулями, использующими методы искусственного интеллекта, что позволяет проводить комплексную цифровую трансформацию образовательного процесса. Разработаны подходы к интеграции интеллектуальных модулей и различных цифровых образовательных сред, что обеспечивает гибкость и масштабируемость образовательной экосистемы;

2. Разработан нейросетевой модуль проверки свободных текстовых ответов обучающихся, основанный на семантической проверке с использованием построения и обработки графов синтаксического разбора предложения, с учетом синонимов, антонимов, структуры предложений и многоуровневых отрицаний, что позволяет объективно оценивать смысловое содержание развёрнутых ответов и существенно снижать нагрузку на преподавателей;

3. Разработаны нейросетевые ассистенты для контекстной помощи обучающимся в процессе решения задач, способствуя развитию навыков самостоятельной работы, а также для помощи преподавателям при создании обучающих материалов и проведении занятий, что оптимизирует методическую работу и повышает вариативность учебных материалов;

4. Разработан модуль поведенческой аналитики и адаптивного обучения, включающий систему сбора и обработки цифрового следа обучающихся, а также метод его представления в виде сигнатур путей, включающий алгоритм обращения сигнатуры и доказанную теорему об оценке погрешности аппроксимации непрерывных траекторий дискретными путями, что позволяет учитывать порядок, взаимосвязи и динамику показателей цифрового следа при построении моделей машинного обучения и обеспечивать генерацию синтетических образовательных траекторий для расширения обучающих выборок.

Полученные в диссертационной работе результаты образуют целостный программно-архитектурный и методический комплекс, в котором архитектура цифровой образовательной платформы, интеллектуальные модули автоматизации и методы анализа цифрового следа функционально взаимосвязаны. Предложенные решения ориентированы на использование в реальных образовательных сценариях и могут быть адаптированы к различным дисциплинам и форматам обучения без изменения базовых архитектурных и алгоритмических принципов.

Практическая направленность диссертационной работы заключается в возможности внедрения разработанных архитектурных решений, интеллектуальных модулей и аналитических методов в состав цифровых образовательных платформ, используемых в образовательных организациях. Предложенные подходы создают основу для дальнейшего развития интеллектуальных средств поддержки обучающихся и преподавателей, расширения аналитических возможностей образовательных систем и применения методов машинного обучения для управления образовательными траекториями.

Таким образом, поставленная цель диссертационного исследования достигнута, все задачи решены. Полученные результаты вносят существенный вклад в развитие отечественных цифровых образовательных технологий и создают основу для дальнейших исследований в области интеллектуальной автоматизации образовательного процесса, адаптивного обучения и анализа образовательных данных.

СПИСОК ЛИТЕРАТУРЫ

1. Аббакумов, А. А. Использование мессенджеров для информирования слушателей учебных заведений / А. А. Аббакумов, Д. П. Сидоров, А. И. Егунова // Образовательные технологии и общество. – 2018. – Т. 21, № 3. – С. 330–336.
2. Башлинская, Ф. Использование программ Microsoft teams и Blackboard в высших образовательных учреждениях / Ф. Башлинская // Norwegian Journal of Development of the International Science. – 2022. – № 82. – С. 39–42.
3. Болтенко, А. П. Особенности применения образовательного портала «ЯКласс» в процессе обучения / А. П. Болтенко // Методика преподавания математических и естественнонаучных дисциплин: современные проблемы и тенденции развития : материалы VI Всероссийской научно-практической конференции. – Омск : ОмГУ, 2019. – С. 106–109.
4. Выбираем LMS. Обзор платформ для дистанционного обучения // vc.ru : [сайт]. – URL: <https://vc.ru/education/163205-vybiraem-lms-obzor-platform-dlya-distancionnogo-obucheniya> (дата обращения: 18.05.2025).
5. Гатулин, Р. Р. Использование мессенджера Telegram для реализации технологии электронного обучения в вузе / Р. Р. Гатулин, Д. А. Колупаева // Санкт-Петербургский образовательный вестник. – 2017. – № 11-12 (15-16). – С. 31–33.
6. Гиматдинова, Г. Н. Возможности использования онлайн-сервиса «ЯКласс» для формирования регулятивных универсальных учебных действий обучающихся 7-9 классов / Г. Н. Гиматдинова // Концепт. – 2020. – № 8. – С. 87–99. – DOI 10.24411/2304-120X-2020-11059.
7. Григорьев, П. В. Особенности технологии RFID и ее применение / П. В. Григорьев // Молодой ученый. – 2016. – № 11(115). – С. 317–322.
8. Григорьев, С. Г. Искусственный интеллект в образовании: приложения систем искусственного интеллекта к анализу и построению онтологических конструкций / С. Г. Григорьев, А. А. Сафронов // Вестник МГПУ. Серия:

Информатика и информатизация образования. – 2024. – № 1(67). – С. 7–17. – DOI 10.25688/2072-9014.2024.67.1.01.

9. Григорьев, С. Г. Искусственный интеллект в образовании / С. Г. Григорьев // Информатизация образования и методика электронного обучения: цифровые технологии в образовании : Материалы VII Международной научной конференции. – Красноярск: Красноярский государственный педагогический университет им. В.П. Астафьева, 2023. – С. 1050–1055.

10. Григорьев, С. Г. Повышение эффективности применения технологий генеративного искусственного интеллекта в образовательной деятельности / С. Г. Григорьев, М. А. Аникьева // Информатика и образование. – 2024. – Т. 39, № 3. – С. 5–15. – DOI 10.32517/0234-0453-2024-39-3-5-15.

11. Данилов, В. В. История создания автоматизированных обучающих систем / В. В. Данилов // Молодой учёный. – 2011. – Т. 2, № 7. – С. 94–98.

12. Задача нахождения объектов на изображении // neerc.ifmo.ru : [сайт]. – URL: https://neerc.ifmo.ru/wiki/index.php?title=Задача_нахождения_объектов_на_изображении (дата обращения: 18.05.2025).

13. Иванова, Е. В. Система учета посещаемости студентов на основе распознавания лиц / Е. В. Иванова, А. Ю. Струева // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2021. – Т. 10, № 4. – С. 60–73. – DOI 10.14529/cmse210404.

14. Ильина, В. Ю. Платформы дистанционного обучения в вузе: альтернативы Moodle и другим зарубежным СДО / В. Ю. Ильина, Ю. М. Кукушкин // Современная филология. Социальная и национальная вариативность языка и литературы : материалы VII Международного научного конгресса. – Симферополь, 2022. – С. 414–419.

15. Кого в бизнесе скоро заменят чат-боты – 5 преимуществ виртуальных помощников // ProBusiness.io : [сайт]. – URL: <https://probusiness.io/tech/3172-kogo-v-biznese-skoro-zamenyat-chat-boty-5-preimushchestv-virtualnykh-pomoshchnikov.html> (дата обращения: 18.05.2025).

16. Кожевников, В. А. Система автоматической проверки ответов на открытые вопросы на русском языке / В. А. Кожевников, О. Ю. Сабинин // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – 2018. – Т. 11, № 3. – С. 57–72. – DOI 10.18721/JCSTCS.11306.
17. Лазарева, О. Ю. Предпосылки возникновения и история развития электронных обучающих систем / О. Ю. Лазарева // Вестник МГУП имени Ивана Федорова. – 2013. – № 9. – С. 76–86.
18. Нурутдинов, А. Р. Перспективы биоинспирированного подхода в разработке систем искусственного интеллекта (обзор тенденций) / А. Р. Нурутдинов, Р. Х. Латыпов // Ученые записки Казанского университета. Серия: Физико-математические науки. – 2022. – Т. 164, № 2-3. – С. 244–265. – DOI 10.26907/2541-7746.2022.2-3.244-265.
19. Онлайн-курсы Eduson: обучение для активных и целеустремленных // Молодой учёный. – 2017. – № 3 (137). – URL: <https://moluch.ru/archive/137/111189/> (дата обращения: 18.05.2025).
20. Открытое образование : [сайт]. – URL: <https://openedu.ru/> (дата обращения: 18.05.2025).
21. Пеккер, П. Л. Измерение эффективности массовых открытых онлайн-курсов: количественные и качественные критерии / П. Л. Пеккер // Высшее образование сегодня. – 2018. – № 8. – С. 37–44. – DOI 10.25586/RNU.NET.18.08.P.37.
22. Применение цифрового следа образовательных интеллектуальных технологий / Л. Ш. Багдасарян, А. Х. Ардеев, Т. А. Куликова, Н. А. Поддубная // Вестник Северо-Кавказского федерального университета. – 2024. – № 2(101). – С. 150–157. – DOI 10.37493/2307-907X.2024.2.18.
23. Рынок чат-ботов в России: итоги 2018 года // comnews.ru : [сайт]. – URL: <https://comnews.ru/digital-economy/content/116845/2019-01-09/rynok-chat-botov-v-rossii-itogi-2018-goda> (дата обращения: 18.05.2025).

24. Система CodeForces : [сайт]. – URL: <https://codeforces.com/> (дата обращения: 18.05.2025).
25. Система ejudge : [сайт]. – URL: <https://ejudge.ru/> (дата обращения: 18.05.2025).
26. Система Яндекс.Контест : [сайт]. – URL: <https://contest.yandex.ru/> (дата обращения: 18.05.2025).
27. Смирнова, О. Г. Опыт использования образовательной платформы Moodle в вузе / О. Г. Смирнова, Е. А. Процукович // The Scientific Heritage. – 2020. – № 51. – С. 46–47.
28. Смирнов, С. Г. Электронный учитель и живой учитель: конкуренты, друзья или сотрудники? / С. Г. Смирнов // Компьютер в школе. – 1999. – № 2. – С. 22–25.
29. Смолянинова, О. Г. Обзор практик обеспечения электронной поддержки образовательного процесса средствами LMS Moodle: опыт российских вузов / О. Г. Смолянинова, Н. А. Иванов // АНИ: педагогика и психология. – 2019. – Т. 8, № 2(27). – С. 228–232. – DOI 10.26140/anip-2019-0802-0054.
30. Среда электронного обучения 3KL Русский Moodle : [сайт]. – URL: <https://opentechnology.ru/products/russianmoodle> (дата обращения: 18.05.2025).
31. Универсариум : [сайт]. – URL: <https://universarium.org/> (дата обращения: 18.05.2025).
32. Фарион, Е. А. Использование электронных образовательных ресурсов / Е. А. Фарион // Pedsovet.su : [сайт]. – URL: <http://pedsovet.su/publ/28-1-0-1051> (дата обращения: 18.05.2025).
33. Фёдорова, Е. М. Возникновение и развитие электронного образования / Е. М. Фёдорова // Современная школа : сборник научных трудов / Комсомольск-на-Амуре : Амурский гуманитарно-педагогический государственный университет, 2021. – С. 32–38.
34. Эквио : [сайт]. – URL: <https://e-queo.com/> (дата обращения: 18.05.2025).

35. 01Математика : [сайт]. – URL: <https://01math.com/> (дата обращения: 18.05.2025).
36. 6 Ways Artificial Intelligence and Chatbots Are Changing Education / Chatbots Magazine : [сайт]. – URL: <https://www.techolac.com/education/6-ways-artificial-intelligence-and-chatbots-are-changing-education/> (дата обращения: 18.05.2025).
37. A Comparative Analysis of ChatBots APIs // ActiveWizards : [сайт]. – URL: <https://activewizards.com/blog/a-comparative-analysis-of-chatbots-apis/> (дата обращения: 18.05.2025).
38. A data-driven market simulator for small data environments / H. Buhler, B. Horvath, T. Lyons [et al.] // arXiv preprint. – 2020. – arXiv:2006.14498.
39. Advantages and Constraints of a Hybrid Model K-12 E-Learning Assistant Chatbot / E. H.-K. Wu, C. Lin, Y. Ou [et al.] // IEEE Access. – 2020. – Vol. 8. – P. 77788–77801. – DOI 10.1109/ACCESS.2020.2988252.
40. Almazroei, E. Automated learning style classification using auto-encoders and Felder-Silverman model from students' digital footprints / E. Almazroei // Machine Intelligence Research. – 2024. – Vol. 18, No. 2. – P. 1–23. – ISSN 2731-538X.
41. Alzahrani, B. Biometric System: Security Challenges and Solutions / B. Alzahrani, F. Alsolami // Proceedings of the 16th International Conference on Information Technology-New Generations (ITNG 2019). – 2019. – P. 111–117. – DOI 10.1007/978-3-030-14070-0_17.
42. Ambati, S. H. Learning AI coding style for software plagiarism detection / S. H. Ambati, N. Stakhanova, E. Branca // Proceedings of the International Conference on Security and Privacy in Communication Systems. – Hong Kong, 2023. – P. 467–489. – DOI 10.1007/978-3-031-64954-7_24.
43. Anderson, C. Creating a Data-Driven Organization : Practical Advice from the Trenches / C. Anderson – 1st ed. – Sebastopol : O'Reilly Media, Inc. – 2015. – 300 P.
44. An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems / Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, P. Martineau // Proceedings of the

4th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2015). – Lisbon, 2015. – P. 271–278. – DOI 10.5220/0005209202710278.

45. Apoki, U. From Digital Learning Resources to Adaptive Learning Objects: An Overview / U. Apoki, H. Al-Chalabi, G. Crisan // Communications in Computer and Information Science. – 2020. – Vol. 1126. – P. 18–32. – DOI 10.1007/978-3-030-39237-6_2.

46. A practical perspective on knowledge space theory: ALEKS and its data / E. Cosyn, H. B. Uzun, C. Doble, J. Matayoshi // Journal of Mathematical Psychology. – 2021. – Vol. 101. – 31 P. – DOI 10.13140/RG.2.2.20082.53448.

47. Arnold, K. Course Signals at Purdue: Using Learning Analytics to Increase Student Success / K. Arnold, M. D. Pistilli // Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. – 2012. – P. 267–270. – DOI 10.1145/2330601.2330666.

48. Azcona, D. Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints / D. Azcona, I. H. Hsiao, A. F. Smeaton // User Modeling and User-Adapted Interaction. – 2019. – Vol. 29. – P. 759–788. – DOI 10.1007/s11257-019-09234-7.

49. Bahel, V. Text similarity analysis for evaluation of descriptive answers / V. Bahel, A. Thomas // arXiv preprint. – 2021. – arXiv:2105.02935.

50. BeGrading: large language models for enhanced feedback in programming education / M. Yousef, K. Mohamed, W. Medhat [et al.] // Neural Computing and Applications – 2024. – Vol. 37, No. 2. – P. 1027–1040. – DOI 10.1007/s00521-024-10449-y.

51. Brusilovsky, P. Adaptive and Intelligent Web-based Educational Systems / P. Brusilovsky, C. Peylo // International Journal of Artificial Intelligence in Education. – 2003. – Vol. 13. – P. 156–169.

52. Callear, D. CAA of Short Non-MCQ Answers / D. Callear, J. Jerrams-Smith, V. Soh // Proceedings of the 5th Computer Assisted Assessment Conference. – Loughborough : Loughborough University, 2001. – P. 1–14.

53. Carbonell, J. AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction / J. Carbonell // IEEE Transactions on Man-Machine Systems. – 1970. – Vol. 11, No. 4. – P. 190–202. – DOI 10.1109/TMMS.1970.299942.
54. Chen, K.-T. Integration of paths, geometric invariants and a generalized Baker-Hausdorff formula / K.-T. Chen // Annals of Mathematics. – 1957. – Vol. 65, No. 1. – P. 163–178. – DOI 10.2307/1969671.
55. Chen, K.-T. Integration of paths – a faithful representation of paths by non-commutative formal power series / K.-T. Chen // Transactions of the American Mathematical Society. – 1958. – Vol. 89. – P. 395–407. – DOI 10.1090/S0002-9947-1958-0106258-0.
56. Chevyrev, I. A Primer on the Signature Method in Machine Learning / I. Chevyrev, A. Kormilitzin // arXiv preprint. – 2016. – arXiv:1603.03788.
57. Chevyrev, I. Signature moments to characterize laws of stochastic processes / I. Chevyrev, H. Oberhauser // arXiv preprint. – 2018. – arXiv:1810.10971.
58. Chrysoulas, C. Building an Adaptive E-learning System / C. Chrysoulas, M. Fasli // Proceedings of the 9th International Conference on Computer Supported Education. – Vol. 2. – Porto, 2017. – P. 375–382. – DOI 10.5220/0006326103750382.
59. Cognitive tutors: Lessons learned / J. R. Anderson, A. T. Corbett, K. R. Koedinger, R. Pelletier // The Journal of the Learning Sciences. – 1995. – Vol. 4, No. 2. – P. 167–207. – DOI 10.1207/s15327809jls0402_2.
60. Conklin, T. A. Knewton / T. A. Conklin // Academy of Management Learning and Education. – 2016. – Vol. 15, No. 3. – P. 635–639. – DOI 10.5465/amle.2016.0206.
61. CoreNLP : [сайт]. – URL: <https://stanfordnlp.github.io/CoreNLP/> (дата обращения: 18.05.2025).
62. Cosma, G. Towards a Definition of Source-Code Plagiarism / G. Cosma, M. Joy // IEEE Transactions on Education. – 2008. – Vol. 51, No. 2. – P. 195–200. – DOI 10.1109/TE.2007.906776.

63. Coursera – доступное онлайн-образование // Навигатор образования : [сайт]. – URL: https://fulledu.ru/articles/1410_coursera-dostupnoe-onlain-obrazovanie.html (дата обращения: 18.05.2025).
64. COVID-19 Academic Integrity Violations and Trends: A Rapid Review / T. Maryon, V. Dubre, K. Elliot [et al.] // Education Sciences. – 2022. – Vol. 12, No. 12. – Art. 901. – DOI 10.3390/educsci12120901.
65. Curtis, G. J. An examination of factors related to plagiarism and a five-year follow-up of plagiarism at an Australian university / G. J. Curtis, R. Popal // International Journal for Educational Integrity. – 2011. – Vol. 7, No. 1. – P. 30–42. – DOI 10.21913/IJEI.v7i1.742.
66. Davar, N. F. AI Chatbots in Education: Challenges and Opportunities / N. F. Davar, M. A. A. Dewan, X. Zhang // Information. – 2025. – Vol. 16, No. 3. – Art. 235. – DOI 10.3390/info16030235.
67. Designing Adaptive Instruction for Teams: a Meta-Analysis / R. Sottolare, C. Burke, E. Salas [et al.] // The International Journal of Artificial Intelligence in Education. – 2017. – Vol. 28, No. 2. – P. 225–264. – DOI 10.1007/s40593-017-0146-z.
68. Development and Evaluation of an Attendance Tracking System Using Smartphones with GPS and NFC / T.-W. Chiang, C.-H. Yang, G.-J. Chiou [et al.] // Applied Artificial Intelligence. – 2022. – Vol. 36, No. 1. – 21 P. – DOI 10.1080/08839514.2022.2083796.
69. Devore-McDonald, B. Mossad: Defeating software plagiarism detection / B. Devore-McDonald, E. D. Berger // Proceedings of the ACM on Programming Languages. – 2020. – Vol. 4, No. OOPSLA. – Art. 138. – DOI 10.1145/3428206.
70. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I. Sutskever, K. Chen [et al.] // arXiv preprint. – 2013. – arXiv:1310.4546.
71. EduChat: A Large-Scale Language Model-based Chatbot System for Intelligent Education / Y. Dan, Z. Lei, Y. Gu [et al.] // arXiv preprint. – 2023. – arXiv:2308.02773.

72. Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado, J. Dean // arXiv preprint. – 2013. – arXiv:1301.3781.
73. Epstein, S. The first book of teaching machines / S. Epstein, B. Epstein // New York: Franklin Watts, Inc. – 1961. – 50 P.
74. Fermanian, A. Embedding and learning with signatures / A. Fermanian // Computational Statistics & Data Analysis. – 2021. – Vol. 157. – Art. 107148. – DOI 10.1016/j.csda.2020.107148.
75. Fletcher, J. D. Shared Mental Models in Support of Adaptive Instruction for Teams Using the GIFT Tutoring Architecture / J. D. Fletcher, R. A. Sottolare // The International Journal of Artificial Intelligence in Education. – 2018. – Vol. 28, No. 2. – P. 265–285. – DOI 10.1007/s40593-017-0147-y.
76. Friz, P. K. A course on rough paths: With an introduction to regularity structures / P. K. Friz, M. Hairer. // 2nd ed. – Berlin : Springer. – 2020. – 346 P. – DOI 10.1007/978-3-030-41556-3.
77. Gao, J. A Similarity Measurement Method Based on Graph Kernel for Disconnected Graphs / J. Gao, J. Gao // Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19) – Macao, 2019. – P. 6430–6431. – DOI 10.24963/ijcai.2019/904.
78. Geolocation API : [сайт]. – URL: https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API/ (дата обращения: 18.05.2025).
79. Gomaa, W. H. A survey of text similarity approaches / W. H. Gomaa, A. A. Fahmy // International Journal of Computer Applications. – 2013. – Vol. 68, No. 13. – P. 13–18. – DOI 10.5120/11638-7118.
80. Gorbatov, S. V. A digital footprint as a mechanism of individualizing a student's educational trajectory (on the case of the "Digital technologies of self-education" course) / S. V. Gorbatov, E. A. Krasnova // Perspectives of Science and Education. – 2022. – Vol. 58, No. 4. – P. 193–208. – DOI 10.32744/pse.2022.4.12.
81. Govindarajan, K. Dynamic Learning Path Prediction – A Learning Analytics Solution / K. Govindarajan, V. Kumar, V. Kumar, Kinshuk // Proceedings

of the 2016 IEEE 8th International Conference on Technology for Education (T4E). – Mumbai, 2016. – P. 188–193. – DOI 10.1109/T4E.2016.047.

82. GPLAG: detection of software plagiarism by program dependence graph analysis / C. Liu, F. Chen, J. Han, P. S. Yu // Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. – Philadelphia, PA, 2006. – P. 872–881. – DOI 10.1145/1150402.1150522.

83. Graham, B. Sparse arrays of signatures for online character recognition / B. Graham // arXiv preprint. – 2013. – arXiv:1308.0371.

84. Graph-based keyword extraction / O. Alqaryouti, H. Khwileh, T. Farouk [et al.] // Intelligent Natural Language Processing: Trends and Applications. – 2018. – P. 159–172. – DOI 10.1007/978-3-319-67056-0_9.

85. GrassBlade LRS : [сайт]. – URL: <https://www.nextsoftwaresolutions.com/> (дата обращения: 18.05.2025).

86. Gyurko, L. G. Extracting information from the signature of a financial data stream / L. G. Gyurko, T. Lyons, M. Kontkowski, J. Field // arXiv preprint. – 2014. – arXiv:1307.7244v2.

87. Hambly, B. Uniqueness for the signature of a path of bounded variation and the reduced path group / B. Hambly, T. Lyons // Annals of Mathematics. – 2010. – Vol. 171, No. 1. – P. 109–167. – DOI 10.4007/annals.2010.171.109.

88. Heift, T. Web Delivery of Adaptive and Interactive Language Tutoring: Revisited / T. Heift, D. Nicholson // International Journal of Artificial Intelligence in Education. – 2016. – Vol. 26, No. 1. – P. 489–503. – DOI 10.1007/s40593-015-0061-0.

89. Holohan, E. Adaptive E-learning content generation based on semantic web technology / E. Holohan, M. Melia, D. McMullen, C. Pahl // Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL 2005). – Amsterdam, 2005. – P. 45–52.

90. Hourrane, O. Rich style embedding for intrinsic plagiarism detection / O. Hourrane // International Journal of Advanced Computer Science and Applications. – 2019. – Vol. 10, No. 11. – P. 646–651. – DOI 10.14569/IJACSA.2019.0101185.

91. Hybrid Approach to Predicting Learning Success Based on Digital Educational History for Timely Identification of At-Risk Students / T. A. Kustitskaya, R. V. Esin, Y. V. Vainshtein, M. V. Noskov // Education Sciences. – 2024. – Vol. 14, No. 6. – P. 657–676. – DOI 10.3390/educsci14060657.
92. Iftakhar, S. Google Classroom: What Works and How? / S. Iftakhar // Journal of Education and Social Sciences. – 2016. – Vol. 3. – P. 12–18. – ISSN 2289-9855.
93. InStudy : [сайт]. – URL: <https://instudy.online/> (дата обращения: 18.05.2025).
94. Introductory programming: a systematic literature review / A. Luxton-Reilly, I. Albluwi, B. A. Becker [et al.] // In Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education. – Larnaca, 2018. – P. 55–106. – DOI 10.1145/3293881.3295779.
95. It's weird that it knows what I want: Usability and interactions with copilot for novice programmers / J. Prather, B. N. Reeves, P. Denny [et al.] // ACM Transactions on Computer-Human Interaction. – 2023. – Vol. 31, No. 1. – Art. 4. – DOI 10.1145/3617367.
96. Janati, S. Context Aware in Adaptive Ubiquitous E-learning System for Adaptation Presentation Content / S. Janati, A. Maach, D. Ghanami // Journal of Theoretical and Applied Information Technology. – 2019. – Vol. 97, No. 16. – P. 4424–4438. – ISSN 1992-8645.
97. Joy, M. Plagiarism in programming assignments / M. Joy, M. Luck // IEEE Transactions on Education. – 1999. – Vol. 42, No. 2. – P. 129–133. – DOI 10.1109/13.762946.
98. Kenter, T. Siamese cbow: Optimizing word embeddings for sentence representations / T. Kenter, A. Borisov, M. De Rijke // arXiv preprint. – 2016. – arXiv:1606.04640.
99. Keuning, H. A systematic literature review of automated feedback generation for programming exercises / H. Keuning, J. Jeuring, B. Heeren // ACM

Transactions on Computing Education (TOCE). – 2018. – Vol. 19, No. 1. – P. 1–43. – DOI 10.1145/3231711.

100. Learning Locker LRS : [сайт]. – URL: <https://github.com/LearningLocker/learninglocker> (дата обращения: 18.05.2025).

101. Le, Q. Distributed representations of sentences and documents / Q. Le, T. Mikolov // Proceedings of the 31st International Conference on Machine Learning. – Beijing, 2014. – Vol. 32. – P. 1188–1196.

102. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals / V. I. Levenshtein // Soviet Physics Doklady. – 1966. – Vol. 10, No. 8. – P. 707–710.

103. LLM agents for education: Advances and applications / Z. Chu, S. Wang, J. Xie [et al.] // arXiv preprint. – 2025. – arXiv:2503.11733.

104. Lyons, T. J. Differential equations driven by rough paths / T. J. Lyons, M. Caruana, T. Levy // Lecture Notes in Mathematics. – 2007. – Vol. 1908. – 126 P. – DOI 10.1007/978-3-540-71285-5.

105. Lyons, T. J. Differential equations driven by rough signals / T. J. Lyons // Revista Matemática Iberoamericana. – 1998. – Vol. 14, No. 2. – P. 215–310. – DOI 10.4171/RMI/240.

106. Marcheggiani, D. Encoding sentences with graph convolutional networks for semantic role labeling / D. Marcheggiani, I. Titov // Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2017) – Copenhagen, 2017. – P. 1506–1515. – DOI 10.18653/v1/d17-1159.

107. Measuring the short text similarity based on semantic and syntactic information / J. Yang, Y. Li, C. Gao, Y. Zhang // Future Generation Computer Systems. – 2021. – Vol. 114. – P. 169–180.

108. Miller, G. A. WordNet: a lexical database for English / G. A. Miller // Communications of the ACM. – 1995. – Vol. 38, No. 11. – P. 39–41. – DOI 10.1145/219717.219748.

109. Modeus : [сайт]. – URL: <https://modeus.custis.ru/> (дата обращения: 18.05.2025).
110. Moltudal, S. H. Adaptive learning technology in primary education: Implications for professional teacher knowledge and classroom management / S. H. Moltudal, R. J. Krumsvik, K. L. Høydal // *Frontiers in Education*. – 2022. – Vol. 7. – Art. 830536. – DOI 10.3389/feduc.2022.830536.
111. Moone, N. Comparing variable fingerprints to indicate plagiarism in code / N. Moone // scripties.uba.uva.nl : [сайт]. – URL: <https://scripties.uba.uva.nl/download?fid=682144> (дата обращения: 18.05.2025).
112. MTCNN : [сайт]. – URL: <https://github.com/timesler/facenet-pytorch/> (дата обращения: 18.05.2025).
113. Murray, W. Cheating in Computer Science / W. Murray // *Ubiquity*. – 2010. – Art. 2. – DOI 10.1145/1865907.1865908.
114. Mussida, P. A computational tool for engineer dropout prediction / P. Mussida, P. L. Lanzi // *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)*. – Tunis, 2022. – P. 1571–1576. – DOI 10.1109/EDUCON52537.2022.9766632.
115. Natasha – набор Python-библиотек для обработки текстов на естественном русском языке : [сайт]. – URL: <https://natasha.github.io/> (дата обращения: 18.05.2025).
116. NetworkX – Python-библиотека для работы с графами и сетями : [сайт]. – URL: <https://networkx.org/> (дата обращения: 18.05.2025).
117. Neural architecture for question answering using a knowledge graph and web corpus / U. Sawant, S. Garg, S. Chakrabarti, G. Ramakrishnan // *Information Retrieval Journal*. – 2019. – Vol. 22, No. 3-4. – P. 324–349. – DOI 10.1007/s10791-018-9348-8.
118. NFC Design for Attendance System in the University / M. Karsen, Y. Kurniawan, C. Cassandra, H. Juwitasary // *International Journal of Mechanical Engineering and Technology*. – 2018. – Vol. 9, No. 6. – P. 566–571. – ISSN 0976-6359.

119. Nowcasting with signature methods / S. N. Cohen, S. Lui, W. Malpass [et al.] // arXiv preprint. – 2023. – arXiv:2305.10256v1.
120. Nye, B. D. AutoTutor and family: A review of 17 years of natural language tutoring / B. D. Nye, A. C. Graesser, X. Hu // International Journal of Artificial Intelligence in Education. – 2014. – Vol. 24, No. 4. – P. 427–469. – ISSN 1560-4306.
121. Open Source Framework to build conversational AI // Rasa Open Source : [сайт]. – URL: <https://rasa.com> (дата обращения: 18.05.2025).
122. Osman, A. H. Graph-Based Text Representation and Matching: A Review of the State of the Art and Future Challenges / A. H. Osman, O. M. Barukub // IEEE Access. – 2020. – Vol. 8. – P. 87562–87583. – DOI 10.1109/ACCESS.2020.2993191.
123. Othman, N. Manhattan Siamese LSTM for Question Retrieval in Community Question Answering / N. Othman, R. Faiz, K. Smaïli // Proceedings of the 18th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2019). – Rhodes, 2019. – P. 661–667. – DOI 10.1007/978-3-030-33246-4_41.
124. Othman, N. Question Answering Passage Retrieval and Re-ranking Using N-grams and SVM / N. Othman, R. Faiz // Computación y Sistemas. – 2016. – Vol. 20, No. 3. – P. 483–494. – DOI 10.13053/CyS-20-3-2470.
125. Pappas, C. The History of eLearning Infographic / C. Pappas // eLearning Industry : [сайт]. – URL: <https://elearningindustry.com/history-of-elearning-infographic> (дата обращения: 18.05.2025).
126. Pawar, A. Calculating the similarity between words and sentences using a lexical database and corpus statistics / A. Pawar, V. Mago // arXiv preprint. – 2018. – arXiv:1802.05667.
127. Persico, D. Use of FirstClass as a Collaborative Learning Environment / D. Persico, S. Manca // Innovations in Education and Training International. – 2000. – Vol. 37, No. 1. – P. 34–41. – DOI 10.1080/135580000362061.
128. Pierce, J. Investigating student plagiarism patterns and correlations to grades / J. Pierce, C. Zilles // Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. – 2017. – P. 471–476. – DOI 10.1145/3017680.3017797.

129. Prechelt, L. Finding plagiarisms among a set of programs with JPlag / L. Prechelt, G. Malpohl, M. Philippsen // Journal of Universal Computer Science. – 2002. – Vol. 8, No. 11. – P. 1016–1038. – DOI 10.3217/jucs-008-11-1016.

130. Programming is hard-or at least it used to be: Educational opportunities and challenges of ai code generation / B. A. Becker, P. Denny, J. Finnie-Ansley [et al.] // Proceedings of the 54th ACM Technical Symposium on Computer Science Education. – Toronto, 2023. – Vol. 1. – P. 500–506. – DOI 10.1145/3545945.3569759.

131. Prompt Problems: A new programming exercise for the generative AI era / P. Denny, J. Leinonen, J. Prather [et al.] // Proceedings of the 55th ACM Technical Symposium on Computer Science Education. – 2024. – Vol. 1. – P. 296–302. – DOI 10.1145/3626252.3630909.

132. Reigeluth, C. M. TICCIT to the Future: Advances in Instructional Theory for CAI / C. M. Reigeluth // Journal of Computer-Based Instruction. – 1979. – Vol. 6, No. 2. – P. 40–46.

133. Reliable plagiarism detection system based on deep learning approaches / M. A. El-Rashidy, R. G. Mohamed, N. A. El-Fishawy, M. A. Shouman // Neural Computing and Applications. – 2022. – Vol. 34, No. 21. – P. 18837–18858. – DOI 10.1007/s00521-022-07486-w.

134. Romero, C. Educational Data Mining: A Review of the State of the Art / C. Romero, S. Ventura // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). – 2010. – Vol. 40, No. 6. – P. 601–618. – DOI 10.1109/TSMCC.2010.2053532.

135. Russakovsky, O. ImageNet Large Scale Visual Recognition Challenge / O. Russakovsky, J. Deng, H. Su [et al.] // International Journal of Computer Vision. – 2015. – Vol. 115, No. 3. – P. 211–252. – DOI 10.1007/s11263-015-0816-y.

136. Rusynonyms – GitHub-репозиторий с синонимами и антонимами русского языка : [сайт]. – URL: https://github.com/ahmados/rusynonyms/blob/main/ru_synonyms/synonyms.py (дата обращения: 18.05.2025).

137. Schleimer, S. Winnowing: local algorithms for document fingerprinting / S. Schleimer, D. S. Wilkerson, A. Aiken // Proceedings of the 2003 ACM SIGMOD international conference on Management of data. – San Diego, 2003. – P. 76–85. – DOI 10.1145/872757.872770.

138. Scorm Cloud LRS : [сайт]. – URL: <https://rusticsoftware.com/products/scorm-cloud/lrs/> (дата обращения: 18.05.2025).

139. Shahiri, A. M. A review on predicting student's performance using data mining techniques / A. M. Shahiri, W. Husain, N. A. Rashid // Procedia Computer Science. – 2015. – Vol. 72. – P. 414–422. – DOI 10.1016/j.procs.2015.12.157.

140. Shebbaz, M. Computer Vision based Attendance Management System / M. Shebbaz, S. M. Ahmed, Z. Alam // YMER Digital. – 2024. – Vol. 23, No. 4. – P. 773–778. – DOI 10.37896/YMER23.04/64.

141. Sherwood, B. The TUTOR Language / B. Sherwood. – 1974.

142. Silva, P., Costa, E. Assessing large language models for automated feedback generation in learning programming problem solving / P. Silva, E. Costa // arXiv preprint. – 2025. – arXiv: 2503.14630.

143. Skillpace : [сайт]. – URL: <https://skillspace.ru/> (дата обращения: 18.05.2025).

144. Sleeman, D. Intelligent tutoring systems / D. Sleeman, J. S. Brown. – Academic Press, New York, 1982. – 345 P.

145. Smith, S. Educational uses of the PLATO computer system / S. Smith, B. Sherwood // Science. – 1976. – Vol. 192, No. 4237. – P. 344–352. – DOI 10.1126/science.769165.

146. Stepik для образовательных организаций : [сайт]. – URL: <https://teach.stepik.org/ru/edu-organizations> (дата обращения: 18.05.2025).

147. Subjective Answers Evaluation Using Machine Learning and Natural Language Processing / F. Bashir, H. Arshad, A. Javed [et al.] // IEEE Access. – 2021. – Vol. 9. – P. 158972–158983. – DOI 10.1109/ACCESS.2021.3130902.

148. Supporting Creation of FAQ Dataset for E-learning Chatbot / Y. Sumikawa, M. Fujiyoshi, H. Hatakeyama, M. Nagai // *Intelligent Decision Technologies*. – 2019. – P. 3–13. – DOI 10.1007/978-981-13-8311-3_1.

149. Systematic review of adaptive learning technology for learning in higher education / J. L. Muñoz, F. M. Ojeda, D. L. Jurado [et al.] // *Eurasian Journal of Educational Research*. – 2022. – Vol. 98. – P. 221–233. – DOI 10.14689/ejer.2022.98.014.

150. The Personality of AI Systems in Education: Experiences with the Watson Tutor, a One-on-One Virtual Tutoring System / S. Afzal, B. Dempsey, C. D'Helon [et al.] // *Childhood Education*. – 2019. – Vol. 95, No. 1. – P. 44–52. – DOI 10.1080/00094056.2019.1565809.

151. The robots are coming: Exploring the implications of OpenAI Codex on introductory programming / J. Finnie-Ansley, P. Denny, B. A. Becker [et al.] // In *Proceedings of the 24th Australasian computing education conference*. – 2022. – P. 10–19. – DOI 10.1145/3511861.3511863.

152. The signature-based model for early detection of sepsis from electronic health records in the intensive care unit / J. Morrill, A. Kormilitzin, A. Nevado-Holgado [et al.] // *Computing in Cardiology*. – 2019. – Vol. 46. – 4 P. – DOI 10.22489/CinC.2019.014.

153. The signature of a rough path: Uniqueness / H. Boedihardjo, X. Geng, T. Lyons, D. Yang // *arXiv preprint*. – 2014. – arXiv:1406.7871.

154. Tosheva, S. Adaptive E-learning System in Secondary Education / S. Tosheva, C. Martinovska // *International Journal of Emerging Technologies in Learning (iJET)*. – 2012. – Vol. 7, No. 1. – P. 36–41. – DOI 10.3991/ijet.v7iS1.1913.

155. Towards robust computerised marking of free-text responses / T. Mitchell, T. Russell, P. Broomhead, N. Aldridge // *Proceedings of the 6th CAA Conference – Loughborough, 2002*. – P. 233–249.

156. Universal Dependency Relations – таблица универсальных синтаксических отношений : [сайт]. – URL: <https://universaldependencies.org/u/dep/> (дата обращения: 18.05.2025).

157. VanLehn, K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems / K. VanLehn // *Educational Psychologist*. – 2011. – Vol. 46, No. 4. – P. 197–221. – DOI 10.1080/00461520.2011.611369.

158. Van Meer, E. PLATO: From Computer-Based Education to Corporate Social Responsibility / E. Van Meer // *Iterations: An Interdisciplinary Journal of Software History*. – 2003. – No. 2.

159. Wang, H. Evaluation of Rocketship Education's use of DreamBox Learning's online mathematics program / H. Wang, K. Woodworth. – Menlo Park : SRI International. – 2011.

160. Wang, K. ChatEd: A Chatbot Leveraging ChatGPT for an Enhanced Learning Experience in Higher Education / K. Wang, J. Ramos, R. Lawrence // *arXiv preprint*. – 2023. – arXiv:2401.00052.

161. Watershed LRS : [сайт]. – URL: <https://www.watershedlrs.com/> (дата обращения: 18.05.2025).

162. Wleklinski, N. Skinner's Teaching Machine and Programmed Learning Theory / N. Wleklinski – 2014.

163. YOLOv5 : [сайт]. – URL: <https://github.com/ultralytics/yolov5/wiki/> (дата обращения: 18.05.2025).

164. Zhang, Y. Understanding bag-of-words model: a statistical framework / Y. Zhang, R. Jin, Z.-H. Zhou // *International Journal of Machine Learning and Cybernetics*. – 2010. – Vol. 1, No. 1-4. – P. 43–52. – DOI 10.1007/s13042-010-0001-0.

165. Zou, H. Regularization and variable selection via the elastic net / H. Zou, T. Hastie // *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. – 2005. – Vol. 67, No. 2. – P. 301–320. – DOI 10.1111/j.1467-9868.2005.00503.x.

ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВА О ГОСУДАРСТВЕННОЙ
РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2025694118

**Программный модуль автоматизированной
семантической проверки открытых текстовых ответов
на русском языке (Модуль семантического анализа
текстовых ответов)**

Правообладатель: *Федеральное государственное автономное
учреждение «Федеральный научный центр Научно-
исследовательский институт системных исследований
Национального исследовательского центра «Курчатовский
институт» (RU)*

Авторы: *Мащенко Кирилл Алексеевич (RU), Леонов Александр
Георгиевич (RU)*



Заявка № 2025693066

Дата поступления 25 ноября 2025 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 03 декабря 2025 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2026666357

**Модуль автоматизированного выявления
заимствований в программном коде с учетом
синтаксических особенностей языков
программирования (Модуль выявления заимствований
в программном коде)**

Правообладатель: *Федеральное государственное автономное
учреждение «Федеральный научный центр Научно-
исследовательский институт системных исследований
Национального исследовательского центра «Курчатовский
институт» (RU)*

Авторы: *Леонов Александр Георгиевич (RU), Мащенко Кирилл
Алексеевич (RU)*

Заявка № 2026661683

Дата поступления 17 апреля 2026 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 01 июня 2026 г.



*Руководитель Федеральной службы
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 00a570e4f7ad38d531b4b8818e75f29506
Владелец **Зубов Юрий Сергеевич**
Действителен с 04.09.2025 по 28.11.2026

Ю.С. Зубов

ПРИЛОЖЕНИЕ Б. АКТЫ О ВНЕДРЕНИИ В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС



Федеральное государственное
бюджетное образовательное учреждение
высшего образования
**РОССИЙСКАЯ АКАДЕМИЯ
НАРОДНОГО ХОЗЯЙСТВА
и ГОСУДАРСТВЕННОЙ СЛУЖБЫ
при ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ**

По месту требования

119571, г. Москва, пр-кт Вернадского, д. 82, стр. 1
тел.: +7 (499) 956-99-99

№ _____
на № _____ от _____

О внедрении в образовательный
процесс

Справка

Федеральное государственное бюджетное образовательное учреждение высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации» (далее – Академия), действуя на основании Соглашения о сотрудничестве от 30.10.2024 с Федеральным государственным автономным учреждением «Федеральный научный центр Научно-исследовательский институт системных исследований Национального исследовательского центра "Курчатовский институт"» (далее – НИЦ «Курчатовский институт» - НИИСИ) осуществил внедрение в образовательный процесс Института открытого образования разработанную в НИЦ «Курчатовский институт» - НИИСИ цифровую образовательную платформу «Мирера» для изучения дисциплин «Управление проектами в ML и ИИ», «Алгоритмирование и программирование», «Анализ данных и технологии работы с использованием Python», «Современные информационные технологии», «Информационные системы в экономике».

Проректор

П.С. Клепинин





Федеральное государственное
образовательное бюджетное
учреждение высшего образования
**«Финансовый университет
при Правительстве
Российской Федерации»
(Финансовый университет)**

СПРАВКА

09.07.2025 № 0253/04.02

Москва

О внедрении в образовательный процесс

Федеральное государственное образовательное бюджетное учреждение высшего образования «Финансовый университет при Правительстве Российской Федерации» (далее – Финансовый университет), действуя на основании Соглашения о сотрудничестве от 15.10.2021 № 01-072/10 с федеральным государственным автономным учреждением «Федеральный научный центр Научно-исследовательский институт системных исследований Национального исследовательского центра "Курчатовский институт"» (далее – НИЦ «Курчатовский институт» – НИИСИ) осуществил внедрение в образовательный процесс Института открытого образования Финансового университета на 2025/2026 учебный год разработанную в НИЦ «Курчатовский институт» – НИИСИ цифровую образовательную платформу «Мирера» для изучения дисциплины «Иностранный язык» по направлению подготовки 38.03.01 «Экономика», образовательные программы «Экономика и финансы» и «Бизнес и корпоративные финансы».

Проректор по учебной
и методической работе



Е.А. Каменева

25816

Автономная некоммерческая организация дополнительного профессионального образования «Институт образовательных технологий»



443001, Самарская область,
г.о.Самара, ул. Ленинская д.302, помещение Н 3, ком. 5-7
ИНН/КПП 6311185348 / 631701001
БИК 043601607
www.inott.ru; тел. 8 (846) 267 03 37; e-mail: info@inott.ru

АКТ О ВНЕДРЕНИИ В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС

Автономная некоммерческая организация дополнительного профессионального образования «Институт образовательных технологий», в дальнейшем именуемое АНО ДПО «Институт образовательных технологий», действуя на основании Договора о совместной деятельности от 19.11.2019 с Федеральным научным центром Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт», далее – НИЦ «Курчатовский институт» - НИИСИ, осуществил внедрение в образовательный процесс АНО ДПО «Институт образовательных технологий» разработанные в НИЦ «Курчатовский институт» - НИИСИ цифровую образовательную платформу «Мирера» в курсах:

- «Формирование ценностного отношения к природе и уважения к традиционным российским духовно-нравственным ценностям в интеграции проектной, экспериментальной и игровой деятельности»;
- «Содержание и технологии в развитии технического творчества детей дошкольного образования в условиях реализации ФГОС»;
- «Развитие интеллектуальных способностей детей старшего дошкольного возраста с использованием технологии СМАРТ-ТРЕНИНГА»;
- «Организация Галереи в образовательной организации для решения задач художественно-эстетического и познавательного развития детей на основе интегративного подхода»;
- «Формирование культуры безопасности и базовых национальных ценностей у детей дошкольного возраста»;
- «Проектирование и организация патриотического воспитания дошкольников на основе приобщения к народной культуре»;
- «Педагогика дополнительного образования детей: по программам инженерного профиля» с 01.09.2022 г. по настоящее время.

Внедрение в образовательный процесс цифровую образовательную среду «ПиктоМир» и цифровую образовательную среду «ПиктоМир-К» в курсах:

«Основы алгоритмизации и программирования для дошкольников в цифровой образовательной среде ПиктоМир»;

«Формирование основ алгоритмизации и программирования у учеников начальной школы в цифровой образовательной среде «ПиктоМир» с 01.09.2022 г. по настоящее время.

В АНО ДПО «Институт образовательных технологий» успешно внедрена современная система обучения, базирующаяся на цифровых инструментах: платформе «Мирера» и образовательных средах «ПиктоМир» и «ПиктоМир-К». ЦОП «Мирера» ориентирована на разработку и проведение курсов, в которых сочетаются онлайн- и оффлайн-технологии проведения учебного процесса, с автоматизированной проверкой правильности и самостоятельности выполненных текущих и контрольных заданий, с коммуникацией педагога с обучающимися в процессе занятий и выполнения задач в реальном времени.

Такой подход к организации учебного процесса продемонстрировал впечатляющую результативность - качество образования существенно возросло, а педагогическая практика обогатилась новыми достижениями. На начало учебного года (01.06.2025) количество обучающихся превысило 20 тысяч человек, что подтверждает эффективность выбранной образовательной модели.

Директор АНО ДПО «Институт
образовательных технологий,
к.и.н.



/Е.Ю.Соловей/

«01» июля 2025 г.

АКТ О ВНЕДРЕНИИ В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС

Федеральное государственное автономное образовательное учреждение высшего образования «Казанский (Приволжский) федеральный университет», далее ФГАОУ ВО КФУ, действуя на основании Соглашения о сотрудничестве от 26.12.2024 с Федеральным научным центром Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт», далее – НИЦ «Курчатовский институт» - НИИСИ, осуществил внедрение в образовательный процесс Центра непрерывного повышения профессионального мастерства педагогических работников Республики Татарстан ФГАОУ ВО КФУ - в курсах «Формирование готовности воспитателей к реализации пропедевтического курса по обучению детей основам алгоритмики и инженерии», «Формирование компетенций учителя начальной школы в реализации курса инженерии для детей», «Методика изучения основ инженерного мышления и искусственного интеллекта в основной школе» цифровую образовательную среду «ПиктоМир», и цифровую образовательную среду «ПиктоМир-К», цифровую образовательную среду «КуМир», цифровую образовательную платформу «Мирера» с 06.02.2025 г. по настоящее время.

Полученный педагогический результат в Центре непрерывного повышения профессионального мастерства педагогических работников Республики Татарстан ФГАОУ ВО КФУ позволяет говорить о высокой эффективности внедренных инновационных форм и методов организации учебного процесса с использованием цифровой образовательной платформы «Мирера», цифровых образовательных сред «ПиктоМир», «ПиктоМир-К», «КуМир», разработки НИЦ «Курчатовский институт» - НИИСИ, что подтверждается значительными достижениями в педагогической практике и существенным повышением качества образовательного процесса.

Директор Центра непрерывного
повышения профессионального
мастерства педагогических
работников Республики
Татарстан ФГАОУ ВО КФУ,
д.э.н., профессор



Р.Ф. Шайхисламов /
«02» июля 2025 г.
М.П.



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ УПРАВЛЕНИЯ»
(ГУУ)**



ИНСТИТУТ ИНФОРМАЦИОННЫХ СИСТЕМ

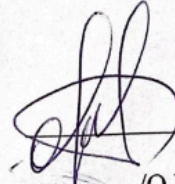
АКТ О ВНЕДРЕНИИ В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС

Федеральное государственное бюджетное образовательное учреждение высшего образования «Государственный университет управления» и именуемое в дальнейшем ФГБОУ ВО ГУУ действуя на основании Договора о сотрудничестве от 10.08.2021 № 34д/03-01-01 с Федеральным научным центром Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт», далее — НИЦ «Курчатовский институт» — НИИСИ, осуществил внедрение в образовательный процесс Института информационных систем (ИИС) по программе «Прикладная математика и информатика» - 01.03.02 разработанные в НИЦ «Курчатовский институт» — НИИСИ цифровую образовательную платформу «Мирера» в курсах: Операционные системы, Алгоритмизация и программирование, Информационные системы, ППП компьютерного моделирования, Курс по глубокому обучению, Обучение с подкреплением, а также цифровую образовательную среду «ПиктоМир» и цифровую образовательную среду «ПиктоМир-К» в курсах: Операционные системы, Алгоритмизация и программирование в период с 01.09.2022 по настоящее время; в образовательный процесс по программе 09.04.03 – «Прикладная информатика» разработанную в НИЦ «Курчатовский институт» — НИИСИ цифровую образовательную платформу «Мирера» в курсах: Методология и технология проектирования информационных систем, Современные технологии разработки программного обеспечения, Математические методы и модели машинного обучения, Компьютерное моделирование и вычислительная математика, Методы программирования и алгоритмы, Временные ряды и их приложения в период с 01.09.2023 по настоящее время.

Полученный педагогический результат в Институте информационных систем ФГБОУ ВО ГУУ позволяет говорить о высокой эффективности внедренных инновационных форм и методов организации учебного с

использованием цифровой образовательной платформы «Мирера», цифровых образовательных сред «ПиктоМир» и «ПиктоМир К», что подтверждается значительными достижениями в педагогической практике и существенным повышением качества образовательного процесса в ИИС ГУУ.

И.о. директора Института
информационных систем ГУУ,
к.э.н., доцент,
зав. кафедрой
математических методов
экономике и управлении



/О.М. Писарева/

«01» июля 2025 г.

М.П.



Подпись

О.М. Писарева

УДОСТОВЕРЯЮ

Зач. Нач. отдела кадров





Министерство просвещения Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский педагогический государственный университет»

улица М. Пироговская, дом 1, строение 1, Москва, 119435

Тел: +7 (499)245-03-10, факс: +7 (499)245-77-58, e-mail: mail@mpgu.su

ОКПО 02079566, ОГРН 1027700215344, ИНН/КПП 7704077771/770401001

64.04.2025 № 074
на № _____ от _____

АКТ О ВНЕДРЕНИИ В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС

В Институте детства федерального государственного бюджетного образовательного учреждения высшего образования «Московский педагогический государственный университет» (далее – МПГУ) внедрены в образовательный процесс по направлениям подготовки 44.03.01 Педагогическое образование, 44.03.02 Психолого-педагогическое образование, 44.04.03 Специальное (дефектологическое) образование, 44.03.05 Педагогическое образование (с двумя профилями подготовки) цифровая образовательная платформа «Мирера», цифровая образовательная среда «ПиктоМир», цифровая образовательная среда «ПиктоМир-К» и цифровая образовательная среда «КуМир», разработанные в Федеральном научном центре Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт» (далее НИЦ «Курчатовский институт» - НИИСИ), в курсах «Алгоритмы и структура данных», «Алгоритмика в начальной школе», «Теория алгоритмов», «Цифровые ресурсы в начальном образовании для детей с ОВЗ», «Воспитание информационной культуры младших школьников», «Теоретические основы информатики», «Архитектура компьютера».

Успешная реализация педагогических инноваций в период с 01.09.2018 г. по настоящее время в Институте Детства МПГУ, основанная на применении современных цифровых инструментов (платформы «Мирера» и образовательных сред «ПиктоМир», «ПиктоМир-К», «КуМир» разработки НИЦ «Курчатовский институт» - НИИСИ), доказала свою результативность: это подтверждается повышением качества подготовки студентов к работе в современном образовательном пространстве.

Директор Института детства



А.А. Алмазова